

# Dokumentation LB1

<b>1. Toolumgebung</b>	2
1.1 Github	2
1.2 Virtualbox	3
1.3 Vagrant	3
1.4 Visual Studio Code	4
<b>2. Infrastruktur</b>	4
2.1 Cloudservice-Modelle	4
2.2 Infrastructure as Code	5
2.3 Vagrant	5
<b>3. Sicherheit</b>	5
3.1 Firewall	5
3.2 Reverse Proxy	6
3.3 Benutzer, Rechte	6
<b>4. Projekt LB1</b>	7
4.1 Vagrantfile	7
4.2 Testfälle	8
4.3 Weiterentwicklungsmöglichkeiten	8
<b>5. Reflexion</b>	9
<b>6. Quellen</b>	9

# 1. Toolumgebung

## 1.1 Github

Account erstellen auf <https://github.com>

Start a Project → M300\_LB1


SSH-Key erstellen:

```
fabri@mssb2c89 MINGW64 ~
$ ssh-keygen -t rsa -b 4096 -c "fabrice.honegger@edu.tbz.ch"
Generating public/private rsa key pair.
Enter file in which to save the key (/c:/Users/fabri/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c:/Users/fabri/.ssh/id_rsa.
Your public key has been saved in /c:/Users/fabri/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:7D5GBInK7Z+H2RCSdchQLCS2k1MoMMO0rvqHeRlVkrE fabri.honegger@edu.tbz.ch
The key's randomart image is:
+---[RSA 4096]-----+
|o=..B*O.|
|..o.o%o+|
|..oo=EB|
|o++o*..|
|..+..=S|
|o o...|
|..B..|
|..=..|
|o..|
+---[SHA256]-----+
```

### SSH-Key registrieren auf Github

SSH keys New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

	<b>SSH_Honigbaer</b> 8e:18:31:22:c1:a5:a4:e1:49:55:e9:d4:e7:fd:b8:86 Added on 21 Mar 2019 Never used — Read/write	<span>Delete</span>
---	--	---------------------

### Git-Client konfigurieren

```
fabri@mssb2c89 MINGW64 ~
$ git config --global user.name "Honigbaer"

fabri@mssb2c89 MINGW64 ~
$ git config --global user.email "fabrice.honegger@edu.tbz.ch"
```

### Repository M300 klonen

```
fabri@mssb2c89 MINGW64 ~
$ git clone https://github.com/mc-b/M300
Cloning into 'M300'...
remote: Enumerating objects: 90, done.
remote: Counting objects: 100% (90/90), done.
remote: Compressing objects: 100% (47/47), done.
remote: Total 542 (delta 36), reused 89 (delta 35), pack-reused 452
Receiving objects: 100% (542/542), 985.62 KiB | 1.12 MiB/s, done.
Resolving deltas: 100% (205/205), done.

fabri@mssb2c89 MINGW64 ~
$ cd m300

fabri@mssb2c89 MINGW64 ~/m300 (master)
$ git pull
Already up to date.

fabri@mssb2c89 MINGW64 ~/m300 (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

## Eigenes Repository erstellen

```
Fabri@MSSB2C89 MINGW64 ~/M300_LB1
$ git clone git@github.com:Honigbaer/M300_LB1.git
Cloning into 'M300_LB1'...
The authenticity of host 'github.com (140.82.118.4)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXupJWG17E1IGOCspromTxdCARLviKw6E5SY8.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added 'github.com,140.82.118.4' (RSA) to the list of known hosts.
Enter passphrase for key '/c/Users/fabri/.ssh/id_rsa':
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.

Fabri@MSSB2C89 MINGW64 ~/M300_LB1
$ git pull
fatal: not a git repository (or any of the parent directories): .git

Fabri@MSSB2C89 MINGW64 ~/M300_LB1
$ ls
M300_LB1/

Fabri@MSSB2C89 MINGW64 ~/M300_LB1
$ cd m300_lb1

Fabri@MSSB2C89 MINGW64 ~/M300_LB1/m300_lb1 (master)
$ git pull
Warning: Permanently added the RSA host key for IP address '140.82.118.3' to the list of known hosts.
Enter passphrase for key '/c/Users/fabri/.ssh/id_rsa':
```

## Repository pushen

```
Fabri@MSSB2C89 MINGW64 ~/M300_LB1/m300_lb1 (master)
$ git add -A .

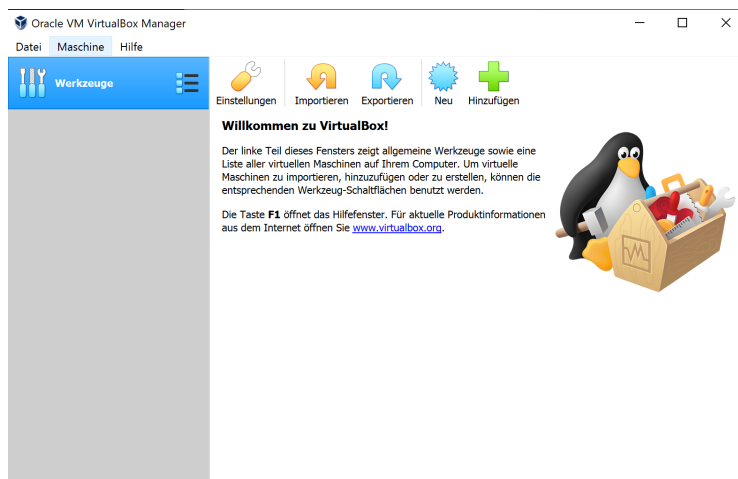
Fabri@MSSB2C89 MINGW64 ~/M300_LB1/m300_lb1 (master)
$ git commit -m "Mein Kommentar"
[master 0b12345] on branch master
1 file changed, 1 insertion(+), 1 deletion(-)
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

Fabri@MSSB2C89 MINGW64 ~/M300_LB1/m300_lb1 (master)
$ git push
Enter passphrase for key '/c/Users/fabri/.ssh/id_rsa':
Everything up-to-date
```

## 1.2 Virtualbox

### Installation Virtualbox

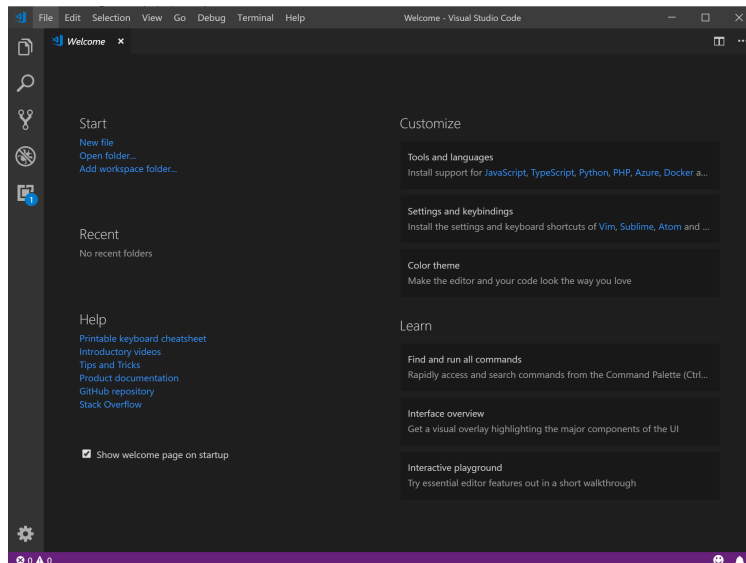


## 1.3 Vagrant

### Installation auf offizieller Webseite

## 1.4 Visual Studio Code

### Installation Visual Studio Code

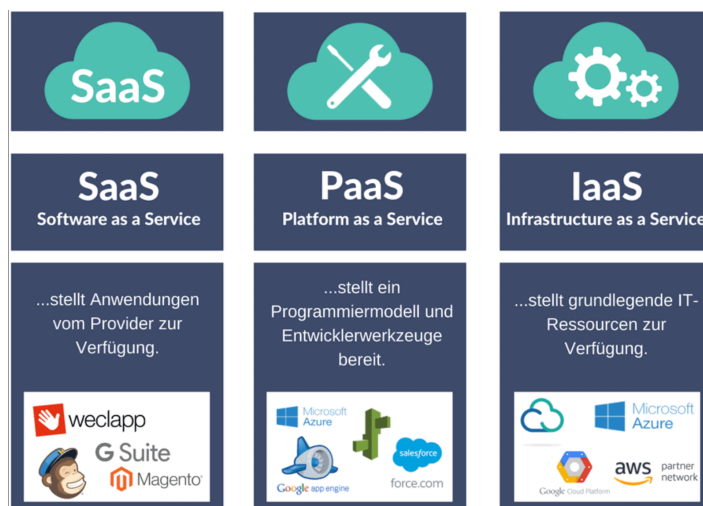


Extension installieren:

- Markdown All in One
- Vagrant Extension
- vscode-pdf Extension

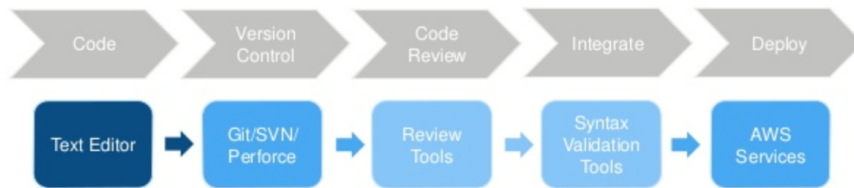
## 2. Infrastruktur

### 2.1 Cloudservice-Modelle



## 2.2 Infrastructure as Code

Prozess bei Infrastructure as Code (IaC)



Voraussetzungen:

- programmierbare API muss bei IaC verfügbar sein.
- on-demand: Maschinen können schnell erstellt und wieder zerstört werden.
- self-service: Ressourcen können selber auf die eigenen Bedürfnisse angepasst werden.
- portabel: Anbieter von Ressourcen (z.B. AWS, Azure) müssen austauschbar sein.

## 2.3 Vagrant

Wichtige Vagrant-Bash-Befehle:

vagrant init	initialisiert Vagrant-Umgebung im Verzeichnis
vagrant up	erzeugt und konfiguriert VM
vagrant ssh	SSH-Verbindung zur VM
vagrant status	zeigt Status der VMs an
vagrant port	zeigt weitergeleitete Ports
vagrant halt	stoppt laufende VM
vagrant destroy	stoppt und entfernt laufende VM
vagrant validate	überprüft vagrantfile auf Lauffähigkeit

Vagrant Boxen:

Webseite: <https://app.vagrantup.com/boxes/search>

Im Vagrantfile kann die gewünschte Box definiert werden und als Grundlage genutzt werden für erweiterte Konfigurationen

Provisioning:

Konfiguration der VM mittels Shell-Befehlen oder anderen Methoden nach der Installation der Box

## 3. Sicherheit

### 3.1 Firewall

Sicherungssystem, welches einen einzelnen Rechner oder ein Netz von Rechnern vor unerwünschten Zugriffen aus dem Netzwerk schützt

Uncomplicated Firewall (UFW) ist ein leistungsstarkes Frontend basierend auf Kommandozeilen

## 3.2 Reverse Proxy

Proxy, der Ressourcen für einen Rechner von einem oder mehreren Servern holt. Die Adresse des Zielsystems bleibt dem Rechner verborgen.

## 3.3 Benutzer, Rechte

Verschiedene Benutzer:

root	Systemverwalter, Administrator
www-data	Benutzer des Webserver Apache
nobody	Benutzererkennung von Prozessen die minimale Rechte benötigen
cupsys	Benutzer des Druckerdienstes CUPS

Datei-/Verzeichnisrechte:

r	Lesen
w	Schreiben
x	Ausführen

chmod	Dateirechte setzen
chown	Dateibesitzer ändern
chgrp	Gruppe einer Datei ändern

## 4. Projekt LB1

### 4.1 Vagrantfile

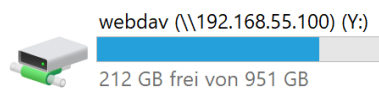
```
1  # -*- mode: ruby -*-
2  # vi: set ft=ruby :
3
4  # Vagrantfile API/syntax version.
5  VAGRANTFILE_API_VERSION = "2"
6
7  Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
8    # All Vagrant configuration is done here. The most common configuration
9    # options are documented and commented below. For a complete reference,
10    # please see the online documentation at vagrantup.com.
11
12    # Konfiguration Virtuelle Maschine Server01
13    config.vm.define "Server01" do |srv01|
14      # verwendete Box für Konfiguration
15      srv01.vm.box = "bento/ubuntu-16.04"
16      srv01.vm.provider "virtualbox" do |vb|
17        # 1024MB Arbeitsspeicher
18        vb.memory = "1024"
19      end
20      # Hostname srv01
21      srv01.vm.hostname = "srv01"
22      srv01.vm.network "private_network", ip: "192.168.55.100"
23      # Einrichtung Share-Ordner
24      srv01.vm.synced_folder ".", "/var/www/html"
25      # Provisionierung, Installation Apache-Webserver, Aktivierung WebDAV-Module, Konfiguration
26      srv01.vm.provision "shell", inline: <<-SHELL
27        sudo apt-get update
28        sudo apt-get -y install apache2
29        sudo ufw allow 'Apache'
30        sudo mkdir /var/www/webdav
31        sudo chown -R www-data:www-data /var/www/webdav
32        sudo a2enmod dav
33        sudo a2enmod dav_fs
34        sudo a2ensite webdav
35        sudo rm /etc/apache2/sites-available/000-default.conf
36        sudo cp /var/www/html/000-default.conf /etc/apache2/sites-available/
37        sudo systemctl restart apache2
38      SHELL
39    end
40    # zusätzliche VM für Testzwecke
41    config.vm.define "Client" do |cl01|
42      cl01.vm.box = "ubuntu/xenial64"
43      cl01.vm.hostname = "cl01"
44      cl01.vm.network "private_network", ip: "192.168.55.101"
45      cl01.vm.provider "virtualbox" do |vb|
46        vb.memory = "1024"
47      end
48      cl01.vm.provision "shell", inline: <<-SHELL
49        sudo apt-get update
50      SHELL
51    end
52  end
53
```


## 4.2 Testfälle

### Dateierstellung lokal auf WebDAV

```
vagrant@srv01:~$ echo "this is a sample text file" | sudo tee -a /var/www/webdav/sample.txt  
this is a sample text file  
vagrant@srv01:~$ cd /var/www/webdav  
vagrant@srv01:/var/www/webdav$ ls  
sample.txt
```

### Einbindung WebDAV als Netzlaufwerk



<input type="checkbox"/> Name	Änderungsdatum	Typ	Größe
 sample.txt	21.03.2019 14:03	Textdokument	1 KB

## 4.3 Weiterentwicklungsmöglichkeiten

- Schreibrechte von extern
- Nutzer und Passwort Erstellung
- Konfigurationsdatei direkt mit Shell-Befehlen anpassen



## 5. Reflexion

Das Thema Vagrant und Docker war vorher noch kein Begriff. Es scheint der logische Schritt in die Zukunft zu sein. Betriebssysteme die man vorkonfigurieren und in einem Schritt aufsetzen und beliebig verbreiten kann. Container bei welchen das Betriebssystem komplett rausgenommen wird und nur noch eine Applikation pro Container läuft.

Auch die Arbeit mit der Linux bash war spannend und noch recht neu für uns.

Etwas mehr Probleme als erwartet hatten wir mit der konkreten Umsetzung des WebDAV-Projekts. Wir mussten einen Weg finden um ein Konfigurationsfile von Apache anzupassen für den WebDAV-Service (via Shell oder das File ersetzen). Die Schreibrechte von extern konnten bis jetzt nicht gegeben werden und müssten in weiterer Entwicklungsarbeit angelegt werden.

## 6. Quellen

Modul 300:

<https://github.com/mc-b/M300>

WebDAV Setup:

<https://www.digitalocean.com/community/tutorials/how-to-configure-webdav-access-with-apache-on-ubuntu-14-04>