> M300 LB02 Dokuemntation - Webserver - Levin Kuhn

Modul 300 Dokumentation

> **Inhaltsverzeichnis**

- Modul 300 Dokumentation
- 1 Einführung
  - 1.1 Einleitung
- 2 Technische Doku
  - 2.1 GIT
  - 2.2 Vagrant
    - 2.3 Vagrantfile
    - 2.4 Vagrantfile Codedokumantation VM
    - 2.5 Vagrantfile Codedokumantation Apache 2
- 3 Tests
- 4 nützliche und verwendete links

1 Einführung

1.1 Einleitung

Das hier ist die Dokuemntation des Moduls 300, in welchem wir einen Webserver automatisieren werden. Zur Realisierung werden Git und Markdown wesentlich beitragen und auch hier in der Dokumentation beschrieben werden. Zum einen wird der Code beschrieben, mit welchem wir alles aufegsetzt haben und zum anderen werden auch Fortschritte hier verzeichnet.

2 Technische Doku

2.1 GIT

Als aller erstes haben wir ein Git Repository erstellt und dies geklont. Um das zu tun habe ich die Anelitung unter folgendem Link verwendet: https://github.com/mc-b/M300/tree/master/10-Toolumgebung

Nun habe ich das Repository lokal geklont und bin in der Lage, meine lokale Arbeit via ***git push*** ins Repository hoch zu laden. Dafür muss man im CMD zum Verzeichnis wechseln in welchem das Repository geklont wurde. Jetzt kann ich den befehl ***git status*** ausführen. Nun sehe ich alle Dateien, in welchen etwas geändert wurde. Mit dem Befehl ***git add -A*** kann ich nun alle Dateien zum Upload bereitstellen. Führt man jetzt erneut einen ***git status*** aus so sind alle vorhin rot markierten Dateien nun grün, was heisst, dass der Upload nun gestartet werden kann. Danach kann der Befehl ***git commit -m "Mein Kommentar"*** ausgeführt werden. Dieser "erlaubt" den Upload und hinterlässt einen Kommentar, welchen den neuen oder geänderten Inhalt des Dokuemnts gut beschreiben sollte. An dieser stelle habe ich erneut einen ***git status***

gemacht, welcher mir nun anzeigt, dass Dateien zum pushen bereit sind. Jetzt konnte ich den Befehl ***git push*** ausführen und auf meinem Repository sehen, dass sich was geändert hat.

> Ergänzung: falls mehrere Branches verwendet werden -> git checkout -> git push -u origin "branchname"

2.2 Vagrant

2.3 Vagrantfile

```
Vagrant.configure("2") do |config|

 config.vm.box = "ubuntu/trusty64"


 config.vm.network "forwarded_port", guest: 80, host: 8080

 # config.vm.synced_folder "../data", "/vagrant_data"

 # Provider-specific configuration s

 config.vm.provider "virtualbox" do |vb|

 vb.gui = true

 vb.memory = "4096"
 end

 config.vm.provision "shell", inline: <<-SHELL
 apt-get update
 apt-get install -y apache2
 SHELL
 end
```

2.4 Vagrantfile Codedokumantation VM

```
  config.vm.box = "ubuntu/trusty64"
```

> *hier ist definiert, welche Image / Box verwendet werden soll*

```
  config.vm.network "forwarded_port", guest: 80, host: 8080
```

> *hier werden die geöffneten Ports eingetragen, damit man später auf die VM zugreifen kann*

```
config.vm.provider "virtualbox" do |vb|
```

> *hier wird definiert welcher Provider verwendet werden soll*

```
vb.gui = true
```

> *hier wird definiert ob ein GUI angezeigt werden soll, bei mir JA*

```
vb.memory = "4096"
```

> *hier wird definiert wie viel Speicher die VM haben soll*

```
config.vm.provision "shell", inline: <<-SHELL
```

> *Diese Zeile ist das Schlusslicht und alles wass noch folgt wird erst nach dem Start der VM ausgeführt*

2.5 Vagrantfile Codedokumantation Apache 2

```
apt-get update
```

> *durch diesen Befehl werden Paketlisten neu eingelesen und aktualisiert*

```
apt-get install -y apache2
```

> *durch diesen Befehl wird apache2 installiert*

- Nun kann man die IP der VM 127.0.0.1 im Browser eingeben. Wen die Apchache2 Standardseite erscheint hat alles funktioniert

> *Anschliessend wurde noch folgenede Befehl ausgeführt*

```
sudo nano /etc/apache2/apache2.conf
```

> *durch diesen Befehl werden wir in das Konfig File kommen, wo wir der Servernamen zu localhost ändern können*

```
sudo service apache2 restart
```

> *durch diesen Befehl könnne wir apache neustarten um die Änderungen zu übernehmen*

```
sudo apache2ctl configtest
```

> *durch diesen Befehl testen wir unser Config. Wir sollte eine Meldung bekommen: Syntax = OK*

3 Tests

Test	Beschreib	Auswertung
1	Nur Vm erstellen	geklappt
2	Konfigurierte VM erstellen	geklappt
3	Komplettes Vagrant UP	geklappt

4 nützliche und verwendete links

- https://github.com/Levo092208/Modul300_LB "Mein Repo"
- https://app.vagrantup.com/boxes/search?utf8=✓&sort=downloads&provider=&q=ubuntu%2Ftrusty64 "UbuntuBox"
- https://bscw.tbz.ch/bscw/bscw.cgi/d32655651/M300_LB2_IaC_1_4.pdf "LB2 Anforderungen"
- https://stackoverflow.com/questions/4181861/message-src-refspec-master-does-not-match-any-when-pushing-commits-in-git "Stackflow Lösungsvorschläge"