

Anleitung zur Aufsetzung eines Lora- und Webservers

1. Installation des Loraservers unter Linux

Der Anleitung auf <https://www.loraserver.io> folgen und die folgenden Abhängigkeiten installieren:

- `sudo apt-get install mosquitto`
- `sudo apt-get install postgresql`
- `sudo apt-get install redis-server`

Datenbank und Schlüssel wie unter <https://www.loraserver.io/loraserver/install/debian/> erstellen und den Server installieren:

- `sudo apt-get install loraserver`

Anschließend den App Server, auch hier muss das SQL Account angepasst werden

<https://www.loraserver.io/lora-app-server/install/debian/>:

- `sudo apt-get install lora-app-server`

Am Ende das Gateway <https://www.loraserver.io/lora-gateway-bridge/install/debian/>:

- `sudo apt-get install lora-gateway-bridge`

Anpassen der Config-Dateien unter:

- `/etc/[loraserver|lora-app-server|lora-gateway-bridge]/[loraserver|lora-app-server|lora-gateway-bridge].toml`

Starten der Programme:

- `sudo systemctl [start|stop|restart|status] [loraserver|lora-app-server|lora-gateway-bridge]`

2. Easy RSA Schlüssel/Certifikat erstellen für die VPN-Verbindung (falls nicht vorhanden)

Easy RSA downloaden und installieren <https://github.com/OpenVPN/easy-rsa>

Im Order `../easy-rsa-3.0.6/easyrsa3/`

- `./easyrsa init-pki`
- `./easyrsa build-ca`

Der Anleitung <https://github.com/OpenVPN/easy-rsa/blob/v3.0.6/doc/EasyRSA-Readme.md#creating-an-easy-rsa-pki> folgen:

- `./easyrsa gen-req ...`

Das Zerifikat an michael.niemetz@oth-regensburg.de (Verwalter des Loragateways)

Starten der VPN Verbindung:

- `sudo openvpn --config ../openvpn/BioTonne_DB0HSR_LoRa_openvpn.conf`

3. Webserver installieren

- `sudo apt-get install apache2`
- `sudo apt install php-fpm`
- `sudo apt-get install php5 libapache2-mod-php5`
- `sudo apt-get install mysql-server mysql-client`

Der Firewallausnahmen hinzufügen:

- `sudo ufw allow 'Apache'`

MySQL setup:

- `sudo mysql`
- `sudo mysql -e "CREATE USER 'username'@'localhost' IDENTIFIED BY 'password';"`
- `sudo mysql -e "GRANT ALL PRIVILEGES ON * . * TO 'username'@'localhost' WITH GRANT OPTION;"`
- `sudo mysql -e "FLUSH PRIVILEGES;"`
- `mysql -u username -p`

Phpmyadmin Installation nach <https://docs.phpmyadmin.net/de/latest/>

- `wget https://www.phpmyadmin.net/downloads/phpMyAdmin-latest-all-languages.zip`
- `unzip phpMyAdmin-latest-all-languages.zip`
- `sudo mkdir /var/www/phpmyadmin`
- `sudo cp -r phpMyAdmin-*/ */var/www/phpmyadmin/`
- `sudo chown -R www-data:www-data /var/www/phpmyadmin`
- `sudo chmod -R 755 /var/www/phpmyadmin`
- `sudo nano /etc/apache2/conf-available/phpmyadmin.conf`
- `sudo a2enconf phpmyadmin`
- `sudo systemctl reload apache2`
- `sudo nano /var/www/phpmyadmin/.htaccess`
- `sudo htpasswd -c /var/www/phpmyadmin/.htpasswd wim44942`

4. Python und Abhängigkeiten

- `sudo apt-get install python3`
- `sudo apt-get install python-mysqldb`
- `pip install mosquito`
- `pip install paho-mqtt`

Python Idle:

- `sudo apt-get install idle3`

5. Benutzerrechte anpassen

- `sudo [chown | chmod] ...+... var/www/[html | phpMyAdmin]`

6. Java-Tool von Vadim

- `sudo apt install maven`
- `sudo apt-get install oracle-java11-installer`
- `sudo apt-get install default-jre`
- `sudo apt-get install default-jdk`

Neu compilieren:

- `cd .../vadim/target/`
- `sudo mvn clean compile assembly:single`

Testen des Codes:

- `java -jar target/dynamic-acs-1.0-jar-with-dependencies.jar --acs --tries 1 --time 5 --localsearch 0 --quiet -p 0 [14,11,10]`