

Raspberry Pi (Edge-Device)

- Das folgende Skript installiert OpenEMS, InfluxDB, Java DK, Grafana, Apache2 auf einem Raspberry Pi
- Getestet mit Debian Bookworm, Trixie - jeweils 64Bit
- Konfiguration für Apache wird erzeugt
- Konfiguration InfluxDB wird erzeugt (admin/admin)
- Die **einzige** Benutzer-Interaktion: folgende Kopieroperationen vom Entwicklungs-System auf den Raspi durchführen:
 - openems.jar -> /usr/share/openems
 - \ui\target\ -> /usr/share/openems/www
 - ggf. c:\openems\config -> /etc/openems.d

```
'clear
'echo "==" : requirements =="
'read -p "press ENTER to continue ..."

'sudo apt install -y wget apt-transport-https gpg
'sudo mkdir -p /etc/apt/keyrings/
'sudo mkdir -p /usr/lib/openems
'sudo mkdir -p /etc/openems.d
'sudo mkdir -p /usr/share/openems
'sudo mkdir -p /usr/share/openems/www
'sudo mkdir -p /var/log/apache2/openems
'sudo chmod -R 750 /var/log/apache2/openems/
'sudo chown -R www-data:www-data /var/log/apache2/openems

'echo "==" : installing sources for Java, Grafana, InfluxDB =="
'read -p "press ENTER to continue ..."

'wget -qO - https://packages.adoptium.net/artifactory/api/gpg/key/public | sudo gpg --dearmor -o /etc/apt/trusted.gpg
'curl -fsSL https://repos.influxdata.com/influxdata-archive_compat.key | gpg --dearmor | sudo tee /etc/apt/trusted.gpg
'wget -q -O - https://packages.grafana.com/gpg.key | sudo gpg --dearmor -o /etc/apt/keyrings/grafana.gpg
'sudo chmod a+r /etc/apt/keyrings/grafana.gpg

'echo "deb [signed-by=/etc/apt/keyrings/grafana.gpg] https://packages.grafana.com/oss/deb stable main" | sudo tee /etc
'echo "deb https://packages.adoptium.net/artifactory/deb $(awk -F= '/^VERSION_CODENAME/{print $2}' /etc/os-release) main" | sudo tee /etc
'source /etc/os-release
'echo "deb [signed-by=/etc/apt/trusted.gpg.d/influxdata.gpg] https://repos.influxdata.com/debian stable main" | sudo tee /etc

'echo "==" : Update repos / install software =="
'read -p "press ENTER to continue ..."

'sudo apt-get update

'sudo apt-get install -y apache2 influxdb grafana temurin-21-jdk mc

'echo "==" : creating config files =="
'read -p "press ENTER to continue ..."

'sudo tee /etc/systemd/system/openems.service > /dev/null << 'EOF'
'[Unit]
'Description=OpenEMS
'After=network.target

'[Service]
'User=root
'Group=root
'iType=notify
'WorkingDirectory=/usr/lib/openems
'LimitCORE=infinity
'LimitRTPRIO=2
'LimitRTTIME=600000000
'CPUSchedulingPolicy=rr
'CPUSchedulingPriority=1
'ExecStart=/usr/bin/java -Dfelix.cm.dir=/etc/openems.d/ -jar /usr/lib/openems/openems.jar
'SuccessExitStatus=143
```

```
Restart=always
RestartSec=10
WatchdogSec=60

[Install]
WantedBy=multi-user.target
EOF

sudo tee /etc/apache2/sites-available/openems.conf > /dev/null << 'EOF'
<VirtualHost *:80>
    #-BASIC-CONFIGURATION
    ServerName localhost
    ServerAdmin admin@localhost
    DocumentRoot "/usr/share/openems/www"

    #-SERVER-CONFIGURATION
    RewriteEngine On

    # REST API
    ProxyPass /rest http://127.0.0.1:8084/rest/
    ProxyPassReverse /rest http://127.0.0.1:8084/rest/

    # WebSocket (corrected)
    ProxyPass /websocket ws://127.0.0.1:8085/
    ProxyPassReverse /websocket ws://127.0.0.1:8085/

    #-DIRECTORY-CONFIGURATION
    <Directory "/usr/share/openems/www">
        RewriteBase /
        RewriteRule ^index\.html$ - [L]
        RewriteCond %{REQUEST_FILENAME} !-f
        RewriteCond %{REQUEST_FILENAME} !-d
        RewriteRule . /index.html [L]
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
    </Directory>

    #-LOGGING
    ErrorLog "/var/log/apache2/openems/error.log"
    CustomLog "/var/log/apache2/openems/access.log" common

    #
    # SetEnvIf Origin "^((http(s)?://[a-zA-Z\-\0-9]+\.\w+\.com(?:\.[a-zA-Z\-\0-9]+)?)$)" CORS_ALLOW_ORIGIN=$1
    # Header always merge Access-Control-Allow-Origin %{CORS_ALLOW_ORIGIN}e env=CORS_ALLOW_ORIGIN
    # Header always merge Access-Control-Allow-Credentials true env=CORS_ALLOW_ORIGIN
    # Header always set Vary "Origin"

</VirtualHost>
EOF

echo "== : enable / configure webserver =="
echo "=== : copy your openems.jar -> /usr/lib/openems/ ==="
echo "=== : copy your /ui/target -> /usr/share/openems/www ==="

read -p "press ENTER to continue ..."

sudo a2enmod rewrite proxy proxy_http proxy_wstunnel headers
sudo a2ensite openems
sudo a2dissite 000-default

echo "== : enable / start services =="
read -p "press ENTER to continue ..."

sudo systemctl daemon-reload
sudo systemctl enable --now influxdb
sudo systemctl enable --now grafana-server
sudo systemctl enable --now openems

sudo systemctl restart apache2

echo "== : create InfluxDB database and admin user =="
read -p "press ENTER to continue ..."

echo "Waiting a few seconds for InfluxDB to be fully up ..."
sleep 5

# Datenbank 'openems' anlegen
influx -execute "CREATE DATABASE openems" || true

# Admin-User 'admin' mit Passwort 'admin' und allen Rechten anlegen
```

```
influx -execute "CREATE USER admin WITH PASSWORD 'admin' WITH ALL PRIVILEGES" || true
# Jetzt Auth aktivieren
sudo sed -i 's/^ *#\? *auth-enabled *= *.* / auth-enabled = true/' /etc/influxdb/influxdb.conf
sleep 5
# InfluxDB mit Auth neu starten
sudo systemctl restart influxdb
echo "InfluxDB setup done: database 'openems' and user 'admin'/'admin' created and auth enabled."...
```