

Deployment

Raspberry Pi

- Leider existiert noch keine detaillierte Anleitung für das Ausrollen des Produktiv-Systems auf einem Raspberry. Die Installation folgte folgenden Schritten:

1. Vorbereitung Raspi:

```
'apt-get install openjdk-8-jre nginx
'curl -sL https://deb.nodesource.com/setup_16.x | sudo -E bash -
'apt-get install -y nodejs
npm install -g @angular/cli
```

2. Einrichtung der Verzeichnisse Raspi:

```
'mkdir /usr/lib/openems
'mkdir /etc/openems.d
'mkdir /usr/share/openems
'mkdir /usr/share/openems/www
```

3. Build der OpenEMS Edge JAR-Datei über Eclipse

- Inside Eclipse IDE open the `io.openems.edge.application` project and open the `EdgeApp.bndrun` file
- Press Export to start the Export Wizard Selection assistant.
- Select Executable JAR and press Next
- Select a Destination for Export to JAR.
- Die erzeugte Java-Datei wird nun per SCP in 's Zielsystem kopiert. Verzeichnis: `/usr/lib/openems/openems.jar`
- Die Konfiguration kann vom Ordner kopiert werden, welcher in `io.openems.edge.application\EdgeApp.bndrun` unter `felix.cm.dir=c:/openems/config` definiert ist. Dieser wird in 's Verzeichnis `/etc/openems.d` kopiert

4. Konfiguration des OpenEMS Edge Dienstes

```
'nano /etc/systemd/system/openems.service
```

mit folgendem Inhalt:

```
'[Unit]
'Description=OpenEMS
'After=network.target
'
'[Service]
'User=root
'Group=root
'Type=notify
'WorkingDirectory=/usr/lib/openems
'LimitCORE=infinity
'LimitRTPRIO=2
'LimitRTTIME=60000000
'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
```

```
systemctl daemon-reload
systemctl enable openems
systemctl restart openems --no-block; journalctl -lfu openems
```

5. UI bauen und installieren: In VisualStudioCode das UI bauen:

```
ng build -c "openems,openems-edge-prod,prod"
```

Die Dateien werden dann im Verzeichnis *B:\eclipse\openems\openems\ui\target* erzeugt und müssen auf den Raspi nach */usr/share/openems/www* kopiert werden

6. Konfiguration nginx mit *nano /etc/nginx/sites-available/openems*

```
server {
    listen 80;
    server_name openems;

    # OpenEMS Web-Interface
    location / {
        root    /usr/share/openems/www;
        index   index.html index.htm;

        error_page   404    300    /index.html;
    }

    # Proxy for OpenEMS REST-API
    location /rest/ {
        proxy_pass      http://127.0.0.1:8084/rest/;

        proxy_set_header    Host $host;
        proxy_set_header    X-Real-IP $remote_addr;
        proxy_set_header    X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header    X-Forwarded-Proto https;
    }

    # Proxy for OpenEMS Websocket
    location /websocket {
        proxy_pass      http://127.0.0.1:8085;

        proxy_http_version    1.1;
        proxy_set_header      Upgrade $http_upgrade;
        proxy_set_header      Connection "upgrade";
    }
}
```

Verlinken und nginx neu starten:

```
ln -s /etc/nginx/sites-available/openems /etc/nginx/sites-enabled/openems
rm /etc/nginx/sites-enabled/default
systemctl nginx restart...
```