

Anker Solix im buanet/iobroker-Container

Das Image `buanet/iobroker` basiert auf **Debian Bookworm** und bringt nur **Python 3.11** mit (`python3` / `python3-dev`). Für diesen Adapter (solixapi) brauchst du **Python 3.12+** sowie die Möglichkeit, ein venv anzulegen.

`PACKAGES=python3-venv python3-pip` reicht **nicht** – das installiert nur Werkzeuge für 3.11, die der Adapter ablehnt.

Zwei praktikable Wege:

Weg	Wann sinnvoll
A – Eigenes Image	Dauerhaft, Updates des Base-Images planbar, kein apt bei jedem Recreate
B – Userscript	Kein Image-Build, schnelle Anpassung am laufenden Stack

Nach beiden Varianten: Adapter installieren und unter **Options** den Python-Befehl auf `/usr/bin/python3.12` setzen (oder leer lassen, wenn die Auto-Erkennung greift). **autoInstallPython** aktivieren bzw. einmal **Python-Abhängigkeiten installieren**.

Offizielle Container-Doku: docs.buanet.de – ioBroker Docker.

Python-3.12-Pakete: [pascallj/python3.12-backport](https://pascallj.com/python3.12-backport) (amd64, arm64, armhf).

A – Eigenes Image (empfohlen)

1. Dateien anlegen

Auf dem Docker-Host einen Ordner anlegen, z. B. `~/iobroker-anker-solix/` :

`Dockerfile`

```
FROM buanet/iobroker:latest

# Python 3.12 neben dem System-Python 3.11 (Bookworm)
# Quelle: https://github.com/pascallj/python3.12-backport
USER root
RUN set -eux; \
    apt-get update; \
    apt-get install -y --no-install-recommends ca-certificates curl; \
    mkdir -p /etc/apt/keyrings; \
    curl -fsSL https://pascalroelev.nl/deb-pascalroelev.gpg \
        -o /etc/apt/keyrings/deb-pascalroelev.gpg; \
    printf '%s\n' \
        'Types: deb' \
        'URIs: http://deb.pascalroelev.nl/python3.12' \
        'Suites: bookworm-backports' \
        'Components: main' \
        'Signed-By: /etc/apt/keyrings/deb-pascalroelev.gpg' \
        > /etc/apt/sources.list.d/pascalroelev.sources; \
    apt-get update; \
    apt-get install -y --no-install-recommends python3.12 python3.12-venv python3.12-dev; \
    python3.12 --version; \
    rm -rf /var/lib/apt/lists/*
```

docker-compose.yml (Beispiel – Volumes/Ports an dein Setup anpassen)

```
services:
  iobroker:
    container_name: iobroker
    build:
      context: .
      dockerfile: Dockerfile
    image: iobroker-anker-solix:latest
    hostname: iobroker
    restart: always
    ports:
      - "8081:8081"
    volumes:
      - iobrokerdata:/opt/iobroker
    environment:
      - TZ=Europe/Berlin

volumes:
  iobrokerdata:
```

Statt `latest` kannst du beim `FROM` und beim Tag eine feste Version nutzen (z.B. `buanet/iobroker:latest-v11`), siehe [Best Practice](#).

2. Bauen und starten

```
cd ~/iobroker-anker-solix
docker compose build --pull
docker compose up -d
```

Bestehenden Container mit gleichem Daten-Volume ersetzen (Backup vorher):

```
docker compose down
docker compose up -d --build
```

3. Prüfen

```
docker exec -it iobroker python3.12 --version
# erwartet z. B.: Python 3.12.x
```

4. Adapter

1. Im Admin: Adapter **anker-solix** installieren bzw. Instanz anlegen.
2. **Options** → **Python-Befehl:** `/usr/bin/python3.12`
3. **Python-Abhängigkeiten installieren** oder Instanz neu starten (`autoInstallPython`).
4. Account / Terms wie in der [README](#) beschreiben.

Manuell im Container (falls nötig):

```
docker exec -it iobroker bash
cd /opt/iobroker/node_modules/iobroker.anker-solix
python3.12 -m venv python/.venv
python/.venv/bin/pip install -r python/requirements.txt
iobroker restart anker-solix.0
```

Image aktualisieren

Bei neuem `buanet/iobroker` -Release erneut bauen:

```
docker compose build --pull
docker compose up -d
```

Die ioBroker-Daten liegen im Volume und bleiben erhalten.

B – Userscript

Ohne eigenes Image: beim Container-Start ein Script ausführen, das Python 3.12 per apt nachzieht. Dafür den Ordner `/opt/userscripts` vom Host mounten.

1. Host-Ordner und Script

Auf dem Host z.B. `~/iobroker-userscripts/` anlegen. Beim **ersten** Start mit leerem Mount legt das Image Beispiel-Dateien an. Danach:

- `userscript_firststart.sh_example` → `userscript_firststart.sh` umbenennen (ohne `_example`)
- Inhalt durch das folgende Script ersetzen (oder die Datei direkt so anlegen)

```
userscript_firststart.sh
```

```
#!/usr/bin/env bash
# Installiert Python 3.12 beim ersten Start eines neuen Containers (buanet/iobroker).
# Läuft erneut nach Recreate des Containers (Container-FS ist dann wieder „frisch“).
set -euo pipefail

MARKER="/opt/.docker_config/.anker-solix-python312"
if [[ -x /usr/bin/python3.12 ]] && /usr/bin/python3.12 --version >/dev/null 2>&1; then
    echo "[anker-solix] Python 3.12 already present: $(/usr/bin/python3.12 --version)"
    exit 0
fi

echo "[anker-solix] Installing Python 3.12 (bookworm backport) ..."
export DEBIAN_FRONTEND=noninteractive
apt-get update
apt-get install -y --no-install-recommends ca-certificates curl
mkdir -p /etc/apt/keyrings
curl -fsSL https://pascalroelevan.nl/deb-pascalroelevan.gpg \
    -o /etc/apt/keyrings/deb-pascalroelevan.gpg
cat >/etc/apt/sources.list.d/pascalroelevan.sources <<'EOF'
Types: deb
URIs: http://deb.pascalroelevan.nl/python3.12
Suites: bookworm-backports
Components: main
Signed-By: /etc/apt/keyrings/deb-pascalroelevan.gpg
EOF
apt-get update
apt-get install -y --no-install-recommends python3.12 python3.12-venv python3.12-dev
python3.12 --version
mkdir -p "$(dirname "$MARKER")"
date -Is >"$MARKER"
echo "[anker-solix] Python 3.12 install finished."
```

Ausführbar machen:

```
chmod +x ~/iobroker-userscripts/userscript_firststart.sh
```

`userscript_everystart.sh` ist für diese Installation **nicht** nötig (apt bei jedem Start wäre langsam). Nur `firststart` reicht: nach Recreate ist das Container-FS neu und `firststart` läuft wieder.

2. Compose / Run anpassen

Volume für Userscripts ergänzen, Image bleibt `buanet/iobroker:...`:

```
services:
  iobroker:
    container_name: iobroker
    image: buanet/iobroker:latest
    hostname: iobroker
    restart: always
    ports:
      - "8081:8081"
    volumes:
      - iobrokerdata:/opt/iobroker
      - ~/iobroker-userscripts:/opt/userscripts
    environment:
      - TZ=Europe/Berlin

volumes:
  iobrokerdata:
```

Container neu erstellen, damit `firststart` greift:

```
docker compose up -d
# bzw. nach Änderung des Scripts / neuen Container:
docker compose down
docker compose up -d
```

Im Container-Log unter dem Schritt zu den Userscripts sollte die Zeile
[anker-solix] Installing Python 3.12 ... bzw. already present erscheinen.

3. Prüfen und Adapter

```
docker exec -it iobroker python3.12 --version
```

Weiter wie unter [A → Adapter](#): **pythonPath** = `/usr/bin/python3.12`, Dependencies installieren.

Hinweise zum Userscript

- Pakete liegen im **Container-Dateisystem**, nicht im ioBroker-Volume. Nach `docker compose down` + neuem Container ohne firststart-Lauf wäre Python wieder weg – deshalb Script aktiv lassen.
- Einfacher **Restart** (`docker restart`) behält die Installation; firststart läuft dann nicht erneut.
- Externe apt-Quelle (Third-Party): nur verwenden, wenn du dem Backport-Repo vertraust. Alternative: eigenes Image mit selbst gebautem Python (aufwändiger).

Nach der Einrichtung – Kurzcheck

Check	Erwartung
<code>python3.12 --version</code> im Container	Python 3.12.x
Adapter-Log	Python setup OK / <code>info.pythonReady</code> = true
Admin Options	<code>pythonPath</code> = <code>/usr/bin/python3.12</code> (empfohlen)

Fehler „Python 3.12+ not found“ → Image/Userscript hat 3.12 nicht installiert oder falscher `pythonPath`.
Fehler zu `venv` / `pip` → Pakete `python3.12-venv` fehlen; Installationsschritt wiederholen.

Fertige Dateien im Repo

Zum Kopieren liegen dieselben Vorlagen unter:

- [docs/docker/Dockerfile] <https://github.com/MatthiasUlrich1/ioBroker.anker-solix/blob/main/docs/docker/Dockerfile>
- [docs/docker/docker-compose.image.yml] <https://github.com/MatthiasUlrich1/ioBroker.anker-solix/blob/main/docs/docker/docker-compose.image.yml>
- [docs/docker/docker-compose.userscript.yml] <https://github.com/MatthiasUlrich1/ioBroker.anker-solix/blob/main/docs/docker/docker-compose.userscript.yml>
- [docs/docker/userscript_firststart.sh] https://github.com/MatthiasUlrich1/ioBroker.anker-solix/blob/main/docs/docker/userscript_firststart.sh