


Docker auf Synology installieren

 **Paket-Zentrum**

<

>

↺

🔍 docker


↓ Installiert 1

📦 Alle Pakete

β Beta-Pakete

👤 Community


Installiert



Docker
03.05.2023

Öffnen

Web Station isntallieren

 **Paket-Zentrum**

<

>

↺


🔍 docker

1 ↓ Installiert

📦 Alle Pakete

β Beta-Pakete

👤 Community



Synology Inc.

Web Station

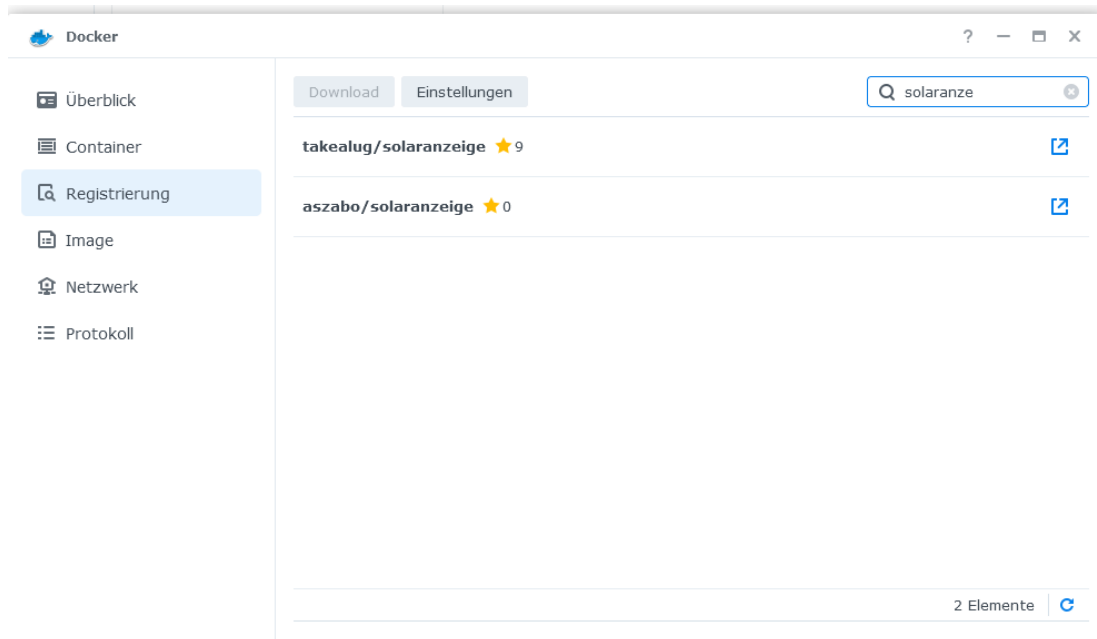
Dienstprogramme

Installieren

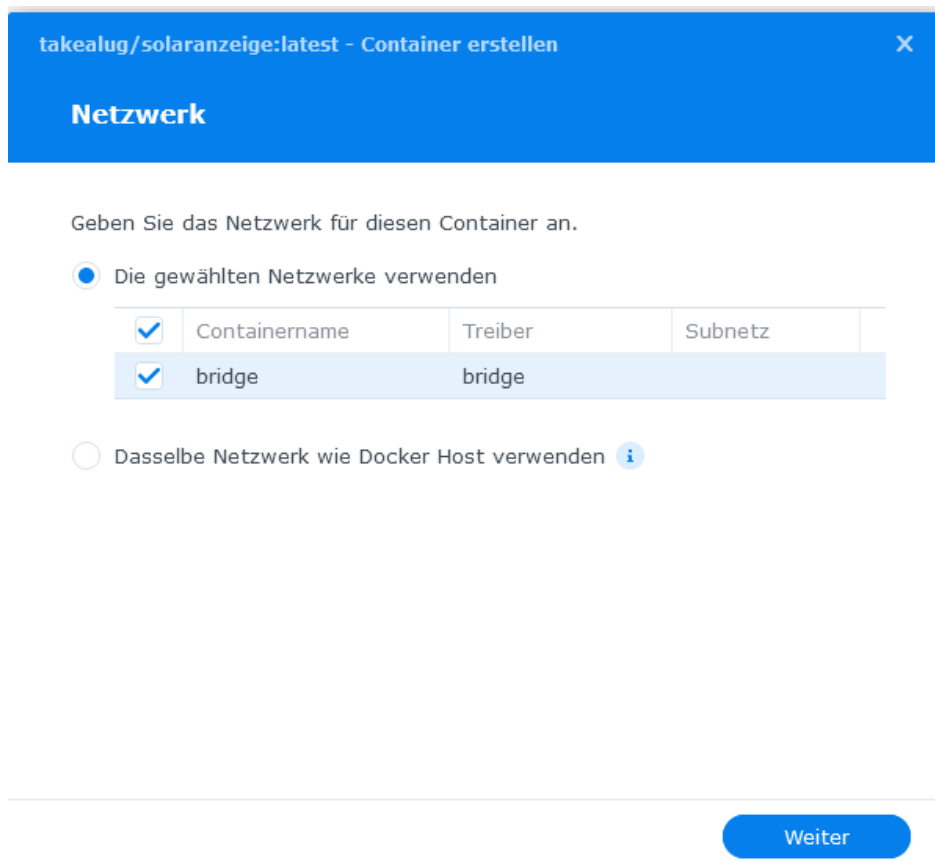
Download Zähler : 8,065,199

Docker öffnen und unter Registrierung nach Solaranzeige suche.

Diese installieren → latest



Danach auf „Starten“ und folgende Einstellungen festlegen



takealug/solaranzeige:latest - Container erstellen

Allgemeine Einstellungen

Containername *:takealug-solaranzeige1

☐ Container mit hoher Priorität ausführen

Funktionen konfigurieren

☐ Ressourcenbeschränkung aktivieren

CPU-PrioritätNiedrig

Speichergrenzwert8192 MB

☒ Automatischen Neustart aktivieren

☒ Webportal via Web Station aktivieren

Port1883

HTTP

Port hinzufügen

Erweiterte Einstellungen

Zurück

Weiter

takealug/solaranzeige:latest - Container erstellen

Port-Einstellungen

Geben Sie verfügbare DSM-Ports in das Feld „Lokaler Port“ ein, um den Ports Container-Ports zuzuordnen. Die hier angeführten Ports sind die offengelegten Ports des Containers.

Hinzufügen

Löschen

Lokaler Port	Container-Port	Typ	
Für Web Station	1883	HTTP	
3000	3000	TCP	
81	80	TCP	

Zurück

Weiter

Die Ordner müssen auf der NAS angelegt und dann hier verknüpft werden.

takealug/solaranzeige:latest - Container erstellen ✕

Volume-Einstellungen

Ordnen Sie die Volumes des Containers gemeinsamen Ordnern auf dem Synology NAS zu.

Datei hinzufügen

Ordner hinzufügen

Löschen

Datei/Ordner	Mount-Pfad	<input type="checkbox"/> S...	:
/docker/Solaranlage/grafana	<input type="text" value="/var/lib/grafana"/>	<input type="checkbox"/>	
/docker/Solaranlage/influxdb	<input type="text" value="/var/lib/influxdb"/>	<input type="checkbox"/>	
/docker/Solaranlage/pvforecast	<input type="text" value="/pvforecast"/>	<input type="checkbox"/>	
/docker/Solaranlage/solaranzeige	<input type="text" value="/solaranzeige"/>	<input type="checkbox"/>	
/docker/Solaranlage/www	<input type="text" value="/var/www"/>	<input type="checkbox"/>	

Zurück

Weiter

Danach den Container starten.

Unter Protokoll warten, bis das Update durch ist.

Dann auf „Terminal“ und „Erstellen“

Hier dann „cd /solaranzeige“ und dann „./setup“ eingeben

takealug-solaranzeige1

Überblick

Prozess

Protokoll

Terminal

Erstellen ▾

Umbenennen

Löschen

takealug-solaranzeige

bash

```
root@takealug-solaranzeige1:/# cd /solaranzeige
root@takealug-solaranzeige1:/solaranzeige# ./setup
```

Mit „W“ und Enter bestätigen.

In der Datei dann die Solaranlage suchen und weiter unten bei „\$Regler =“ eintippen.

Mit F2 speichern und F10 beenden.

Nun den Docker Container nochmal neu starten.

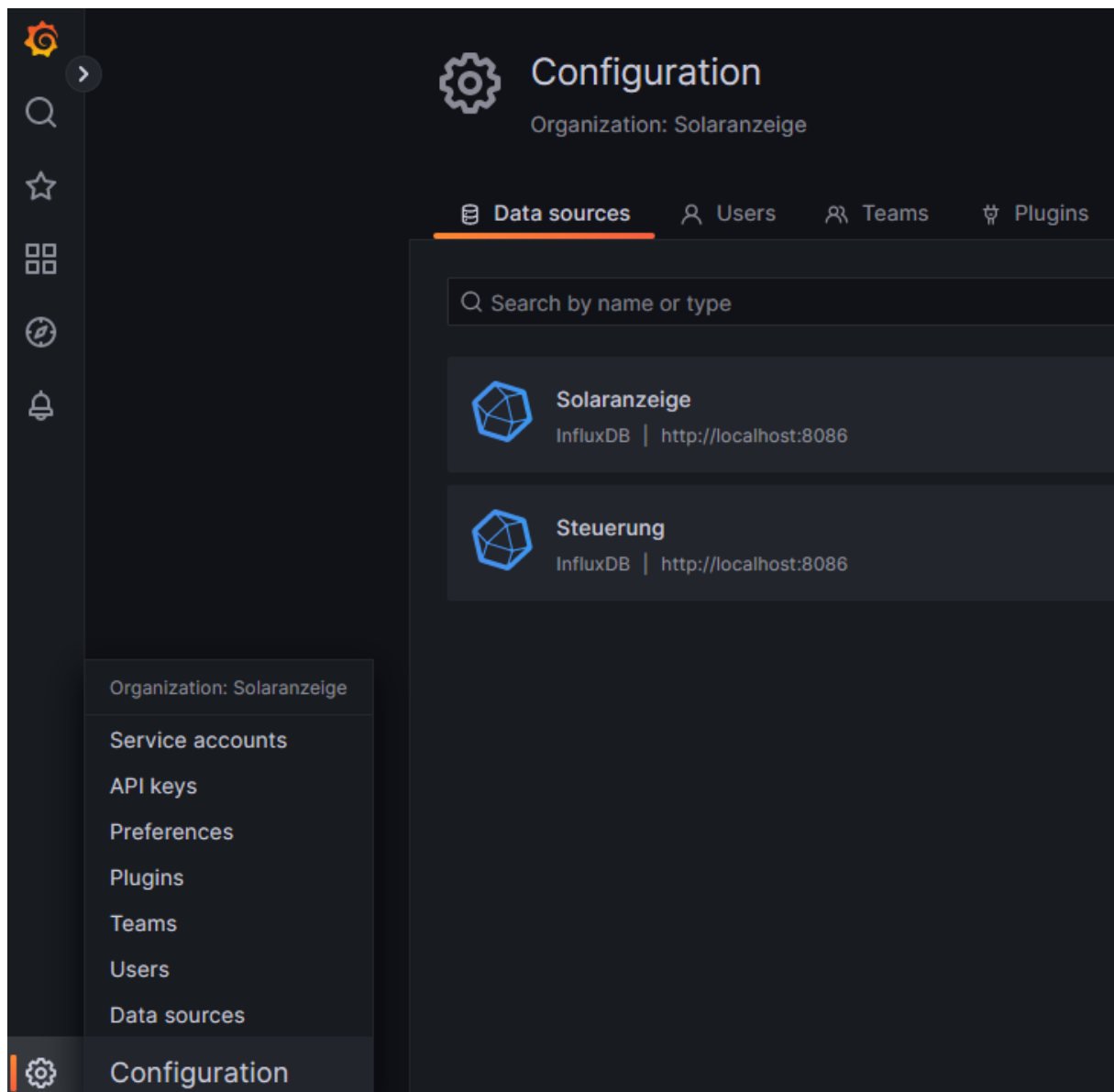
Jetzt sollte das Dashboard unter <http://synology-ip:3000> erreichbar sein.

Hier mit „admin“ und „solaranzeige“ einloggen.

Jetzt das gewünschte Dashboard suchen und anzeigen lassen.

Ich hatte noch das Problem, dass mein Dashboard keine Werte angezeigt hat.

Um das zu lösen musste ich in Grafana unter Configuration und dort Solaranzeige auswählen.




Hier fehlte bei mir ganz unten bei Database der Name.

Diesen habe ich eingetragen und nach einem Klick auf „Save & Test“ und einem Neustart des Containers hat alles geklappt.

Custom HTTP Headers

+ Add header




InfluxDB Details



Database Access

Setting the database for this datasource does not deny access to other databases. The InfluxDB query syntax allow example: `SHOW MEASUREMENTS ON _internal` or `SELECT * FROM "_internal".."database" LIMIT 10`

To support data isolation and security, make sure appropriate permissions are configured in InfluxDB.

Database	solaranzeige	
User	admin	
Password	configured	<div>Reset</div>
HTTP Method	<div> POST</div>	
Min time interval	<div> 30s</div>	
Max series	<div> 1000</div>	

Back

Explore

Delete

Save & test