*Fachhochschule Erfurt*

*Fakultät*

*Gebäudetechnik und Informatik*

*Pflanzennährstoffsteuerung NFT*

Installationshandbuch

*Gültig Ab 04.07.2022*

*PNS Solutions*

*FR Angewandte Informatik*

*Altonaer Str. 25*

*99085 Erfurt*

Inhaltsverzeichnis

[**Installation auf dem Raspberry-Pi**](#_heading=h.30j0zll) **3**

[Verbindung mit dem Raspberry-Pi](#_heading=h.1fob9te) 3

[IP-Adresse des PI’s ermitteln](#_heading=h.mn8cxd1t0u4m) 3

[Update apt-get](#_heading=h.3znysh7) 3

[Git installieren:](#_heading=h.2et92p0) 3

[Programm von GitHub auf den Raspberry-Pi klonen.](#_heading=h.azpflmrcqjk9) 4

[In das geklonte Repository wechseln](#_heading=h.3dy6vkm) 5

[IP-Adresse des Webservers setzen](#_heading=h.1t3h5sf) 6

[Allen Scripten die benötigten Rechte geben](#_heading=h.yq2j2hl7a2cf) 6

[Docker installieren](#_heading=h.4d34og8) 6

[Docker starten](#_heading=h.2s8eyo1) 6

[CronTab hinzufügen](#_heading=h.17dp8vu) 7

[Build Docker](#_heading=h.3rdcrjn) 7

[**Grafana einrichten**](#_heading=h.gyw3791pnidw) **8**

[Datenbank verbinden](#_heading=h.2oohu88a59a2) 8

[Dashboard hinzufügen](#_heading=h.qsn20jlh547m) 10

[**Node-Red einrichten**](#_heading=h.kzbsstmjo4nt) **12**

[Importieren von Flows](#_heading=h.sgfbn8lvkr5w) 12

[Anpassen der IP für den MQTT Broker](#_heading=h.8bgpgimqhcnd) 13

[**SSH-Key Generieren**](#_heading=h.49x2ik5) **15**

[**Sensor- und Pumpen-/Ventileinheit**](#_heading=h.2p2csry) **15**

## Installation auf dem Raspberry-Pi

### Verbindung mit dem Raspberry-Pi

Beim Einrichten eines neuen Servers (Raspberry-Pi) ist die IP-Adresse in der Regel nicht bekannt. Dieser ist aber unter **raspberrypi.local** im Terminal erreichbar sein. Man kann sich also über den folgenden Befehl mit dem PI verbinden:

**ssh USERNAME@raspberrypi.local** (z. B.: ssh pi@raspberrypi.local)

*Auf einen frisch installierten Pi kann nur dann über WiFi zugegriffen werden, wenn die Netzwerkeinrichtung beim Flashen der SD-Karte erfolgte oder es mittels SSH über Ethernet eingerichtet wurde. Für die letztere Option ist folgender Befehl zu verwenden:* ***sudo raspi-config***

*Der Pi kann mittels SSH unter folgendem Befehl aufgerufen werden:*

### IP-Adresse des PI’s ermitteln

Ist man mittels **raspberrypi.local** auf dem PI, kann man über **ifconfig** die IP-Adresse ermitteln. Diese ist unter der Schnittstelle **wlan0** zu finden. Sie ist für die weitere Einrichtung des Systems noch von Bedeutung und sollte notiert werden.

Ist die IP-Adresse bekannt, kann und sollte man sich ab jetzt auch mit dieser über SSH verbinden.

**ssh USERNAME@xxx.xxx.xxx.xxx.xxx** (z. B.: ssh pi@192.168.0.420)

### Update apt-get

In der Konsole des Raspberry-Pi muss folgender Befehl ausgeführt werden:

**sudo apt-get update**

### Git installieren:

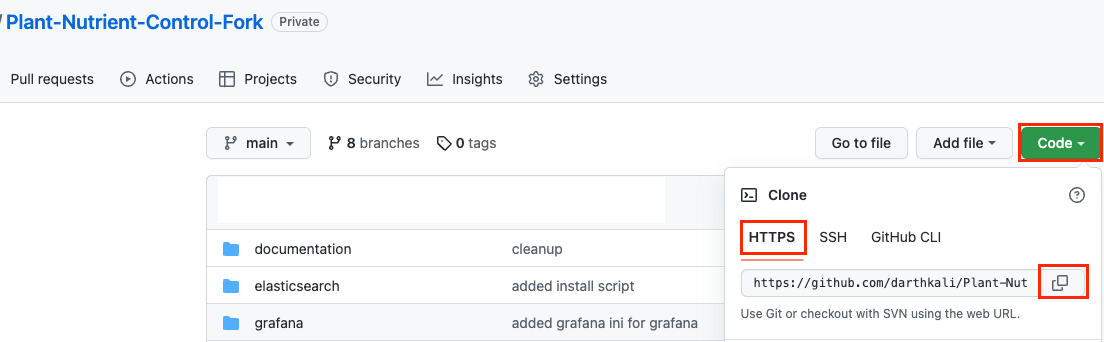
In der Konsole des Raspberry-Pi muss folgender Befehl ausgeführt werden:

**sudo apt install git -y**

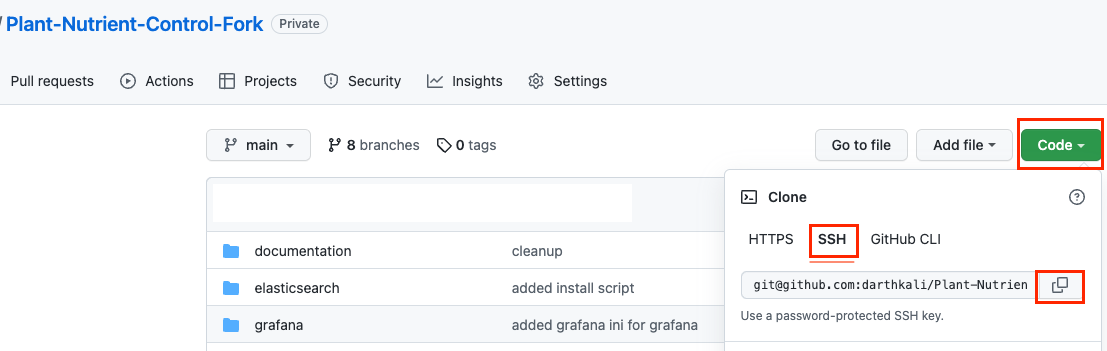
### Programm von GitHub auf den Raspberry-Pi klonen.

Hierfür stehen zwei Möglichkeiten zur Verfügung. Entweder HTTPS (mit einem personal access token) oder SSH.

**HTTPS:**

1. Personal Access Token generieren <https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token>   
     
   *Der Token wird gleich benötigt und kann nicht nochmal generiert werden (sollte er erneut benötigt werden, muss ein neuer Token generiert werden)*
2. Auf das bereitgestellte GitHub Repository gehen und unter HTTPS den Link kopieren.
3. In der Konsole des Raspberry-Pi muss folgender Befehl ausgeführt werden:  
   **git clone *https://github.com/darthkali/Plant-Nutrient-Control-Fork.git*** *(den Link durch den vom GitHub Repository aus Teilschritt 2 ersetzen)*
4. Danach wird nach dem Passwort gefragt. Hier muss nun der im ersten Teilschritt erzeugte Token verwendet werden.

**SSH:**

* 1. (optional) SSH-Key erzeugen: Wenn noch kein SSH-Key vorhanden ist, dann muss dieser erzeugt werden. Wie das geht kann in Abschnitt **SSH-Key Generieren** nachgelesen werden.
  2. Auf das bereitgestellte GitHub Repository gehen und unter SSH den Link kopieren.
  3. In der Konsole des Raspberry-Pi muss folgender Befehl ausgeführt werden:  
     **git clone**[***git@github.com***](mailto:git@github.com)***:darthkali/Plant-Nutrient-Control-Fork.git*** *(den Link durch den vom GitHub Repository aus Teilschritt 2 ersetzen)*

### In das geklonte Repository wechseln

In der Konsole des Raspberry-Pi muss folgender Befehl ausgeführt werden:

**cd Plant-Nutrient-Control-Fork**

Der Name des Ordners kann abweichen. In der Regel sollte es der Name des Repositories sein. Welche Ordner alle im aktuellen Verzeichnis liegen, kann mit dem Befehl

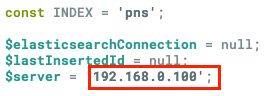
**ls -li**

geprüft werden.

### IP-Adresse des Webservers setzen

In der Konsole des Raspberry-Pi muss folgender Befehl ausgeführt werden:

**nano webapplication/init/10\_database.php**

Dieser öffnet einen Editor in dem die IP-Adresse des Pi’s eingetragen werden kann.****

Nach der Änderung mit **ctrl + s** und **ctrl + x** *(auf Windows Strg + s und Strg + x)* die Datei speichern und den Editor verlassen.

### Allen Scripten die benötigten Rechte geben

Um die Installation so einfach wie möglich zu halten, wurden einige Schritte in Scripte verpackt. Diese benötigt jedoch die Berechtigung auf dem Pi, um ausgeführt zu werden. Diese können mit folgendem Befehl gesetzt werden:

**chmod +x ./scripts/install-docker.sh && chmod +x ./scripts/start-docker.sh && chmod +x ./scripts/build-docker.sh && chmod +x ./scripts/addCronTab.sh**

### Docker installieren

In der Konsole des Raspberry-Pi muss folgender Befehl ausgeführt werden:

**./scripts/install-docker.sh**

Wenn das Script fertig ist, was ein paar Minuten dauern kann, wird der Raspberry Pi neu gestartet. Nach 1–2 Minuten muss sich dann wieder via SSH *(siehe Schritt 2)* auf dem PI eingeloggt werden und mittels **cd** *(siehe Schritt 6)* in das Verzeichnis vom Projekt gewechselt werden.

### Docker starten

In der Konsole des Rasperry-Pi muss folgender Befehl ausgeführt werden:

**./scripts/start-docker.sh**

### CronTab hinzufügen

Damit die Controller (für Sensoren und Pumpen bzw. Ventile) durchgehend laufen, müssen sie gestartet aber vor allem auch kontinuierlich kontrolliert und ggf. neu gestartet werden. Dazu wurde ein CronJob eingerichtet. Dieser wird mit folgendem Befehl gesetzt:

**./scripts/addCronTab.sh**

### Build Docker

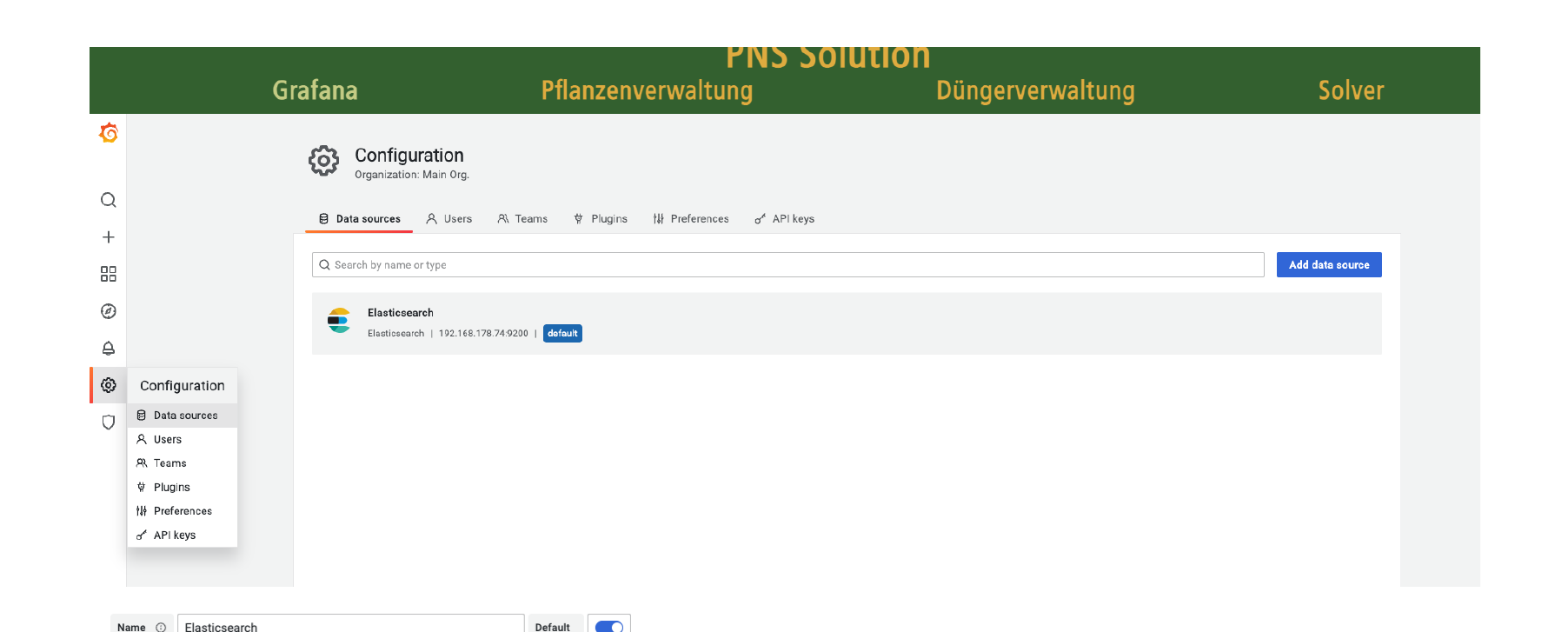
In der Konsole des Raspberry-Pi muss folgender Befehl ausgeführt werden:

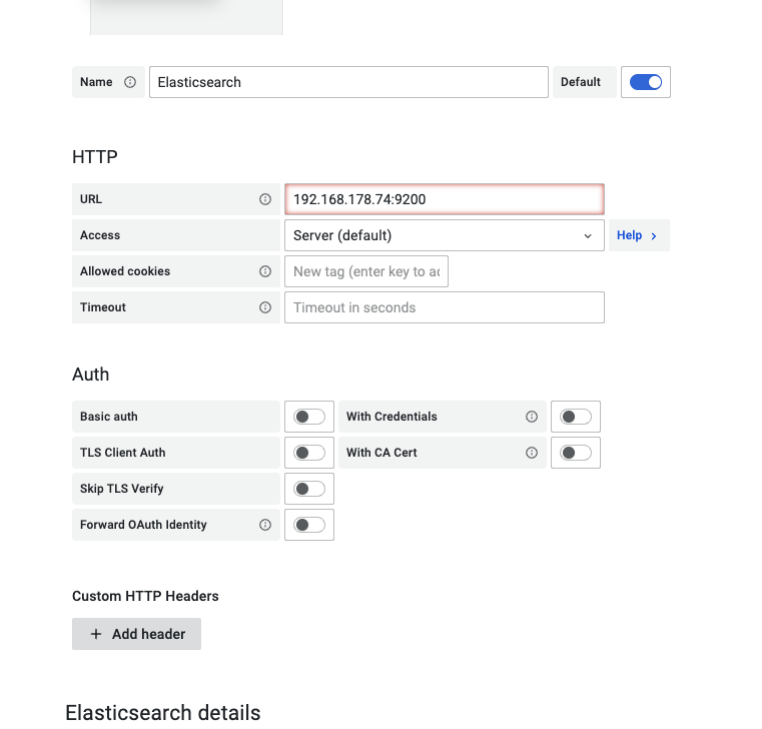
**./scripts/build-docker.sh**

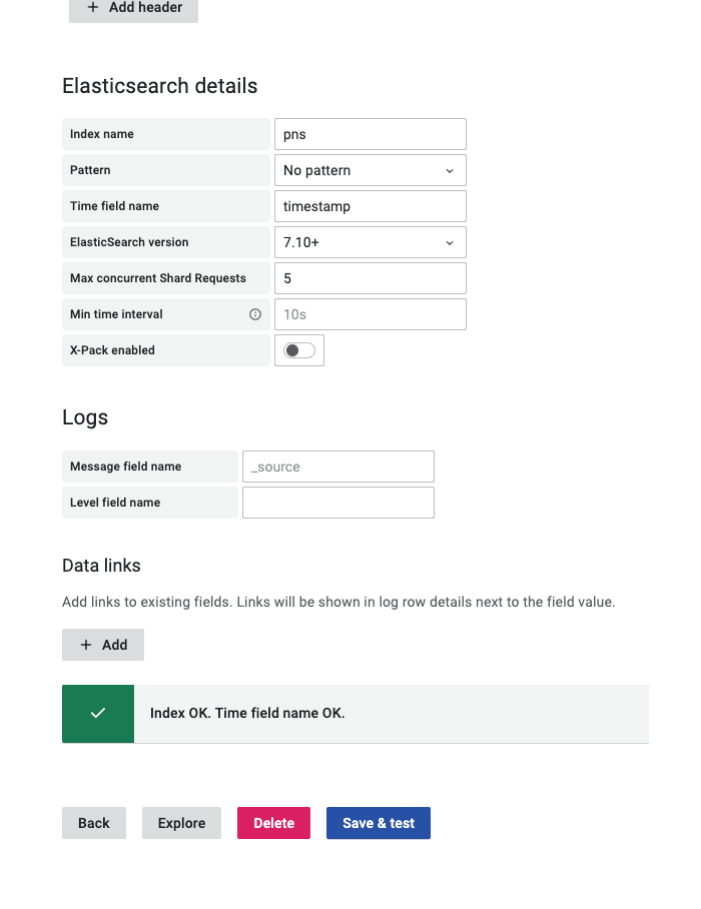
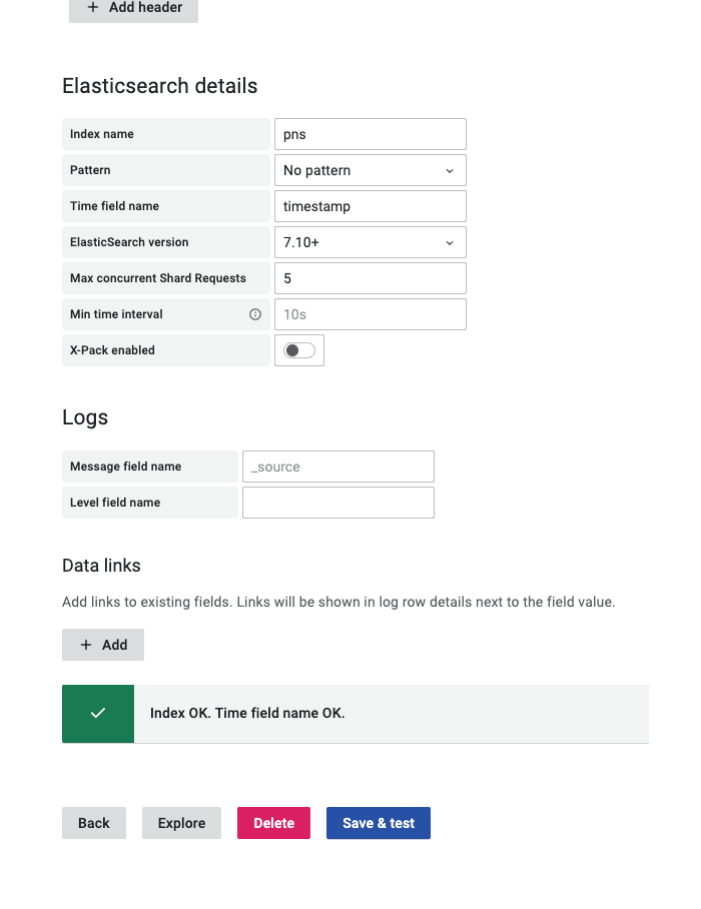
*Damit ist das System installiert und kann nach ein paar Minuten verwendet werden. Die Datenbank braucht in der Regel ein paar Minuten, bis sie betriebsbereit ist.*

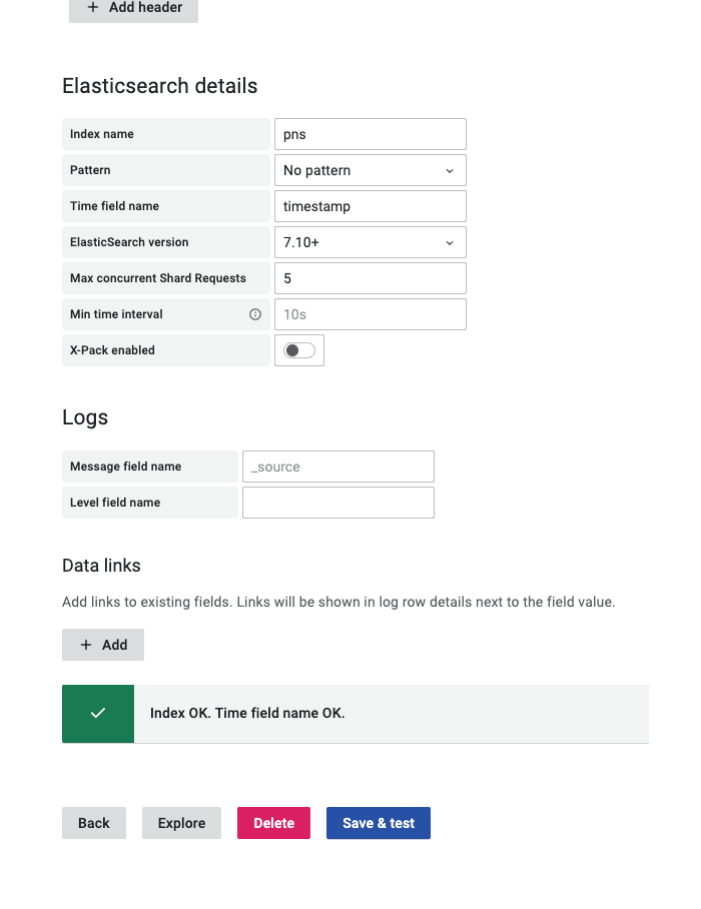
## Grafana einrichten

### Datenbank verbinden

1. Neue DataSource anlegen
2. IP Adresse der DB mit Port eingeben

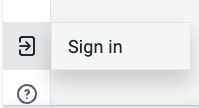


1. Restliche Daten nach dem folgenden Bild eingeben:
2. Auf Save & test klicken

Es sollte die Meldung **Index OK. Time field name OK** erscheinen.

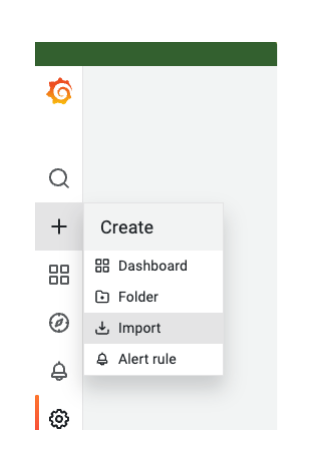
### Dashboard hinzufügen

1. (optional) Um ein Dashboard hinzuzufügen muss man als Admin angemeldet sein. Hierzu unten auf Sign in klicken und anmelden.

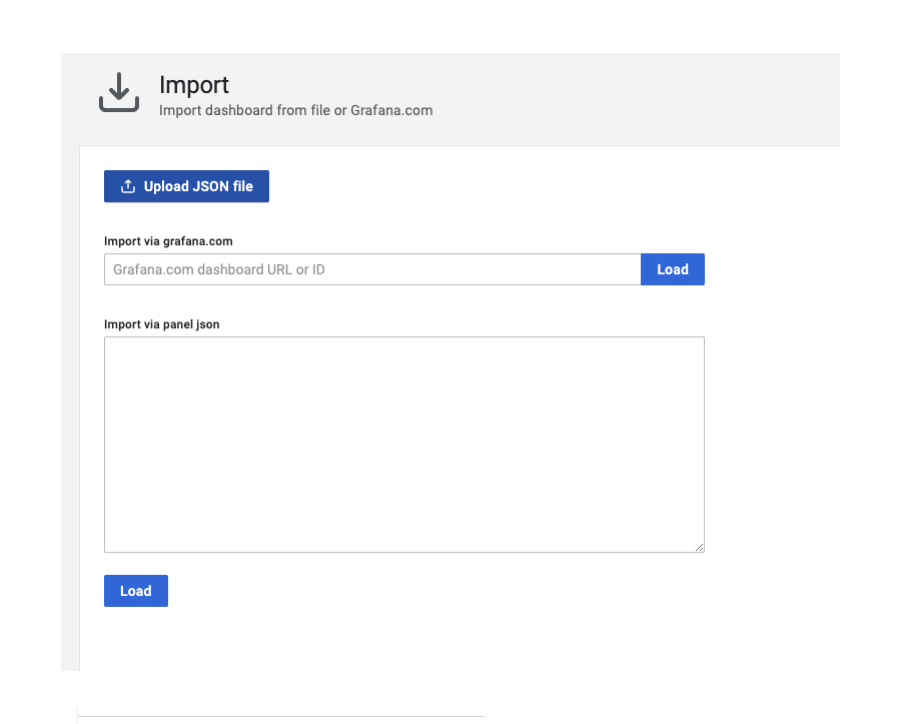


*!! Hinweis: Hierzu ist es unter Umständen nötig, die Seite über die IP und nicht über die Webseite aufzurufen. Dazu einfach in der Adresszeile im Browser die IP vom PI und als Port 3030 eingeben.*

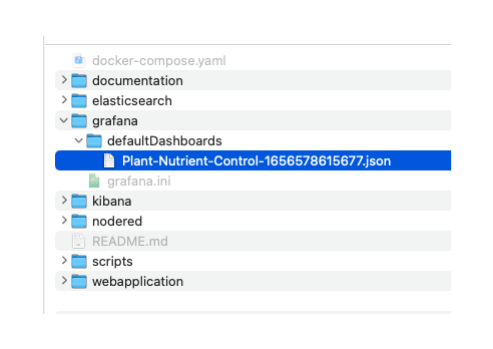
1. Auf der linken Seite über das *Plus-Zeichen* hovern
2. Auf *Import* klicken



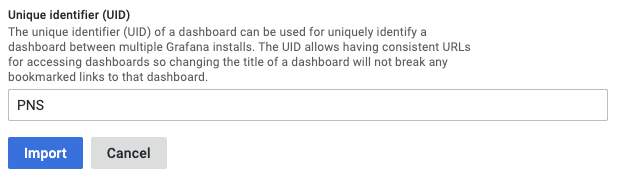
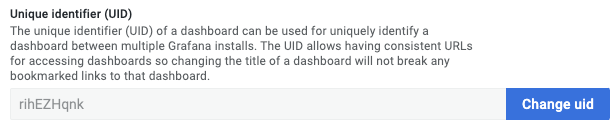
1. Auf *Upload JSON* file klicken



1. JSON aus dem Projektordner laden (grafana -> defaultDashboards)



1. Danach erscheint die UID. Hier auf *Change uid* klicken und als UID **PNS** verwenden.



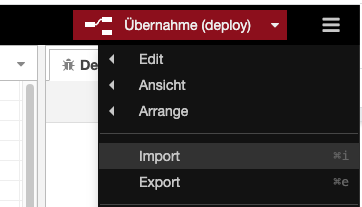
1. Auf *Laden* klicken

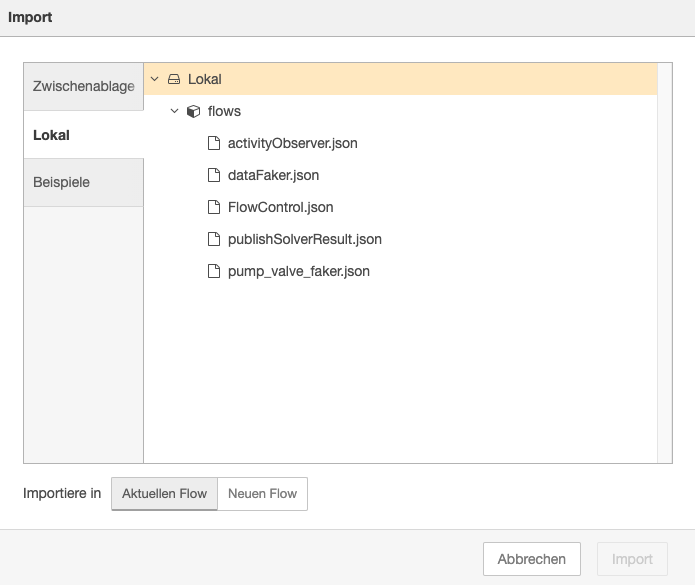
**Hinweis:** Falls die Panels *(im Dashboard wie z.B. der Graph von PH)* nach dem Import des Dashboards nicht angezeigt werden. Muss das Panel ausgewählt werden, und auf *Edit* geklickt werden. Danach rechts am Rand unter „Query inspector“ auf das Augensymbol und dies einmal auf *disable* und wieder auf *enable* stellen.

## Node-Red einrichten

### Importieren von Flows

1. Auf das Menü in der linken oberen Ecke klicken
2. Import wählen



1. Auf den Reiter *Lokal* wechseln
2. Wähle den Flow der importiert werden soll z.B.: *dataFaker.json*
3. Auf *Import* klicken

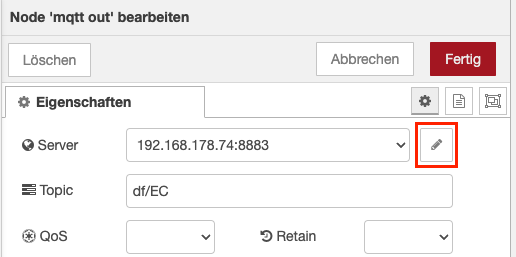
### Anpassen der IP für den MQTT Broker

Damit das MQTT korrekt funktioniert muss der hinterlegte MQTT Server angepasst werden. Dazu folgende Schritte ausführen:

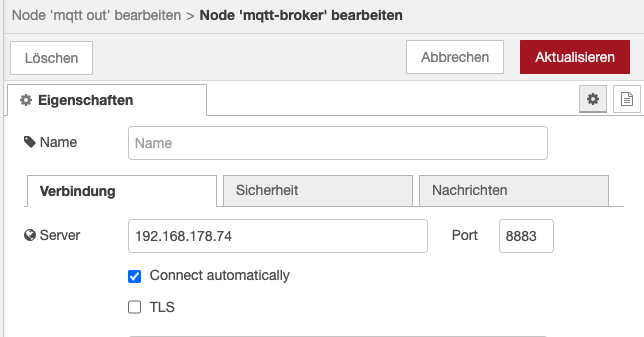
1. Doppelklick auf eines der MQTT-Nodes



1. Server bearbeiten



1. Server IP des Raspberry-Pi angegeben und Port 8883 nutzen



1. Auf *Aktualisieren* klicken
2. Auf *Fertig* klicken
3. Auf *Übernahme (deploy)* klicken



Wenn alles funktioniert hat, sollte unter den MQTT-Nodes jetzt ein Grünes Symbol mit dem Text “*Verbunden*” stehen.



## SSH-Key Generieren

1. In der Konsole des Raspberry-Pi muss folgender Befehl ausgeführt werden:

**ssh-keygen -t rsa**

1. Enter drücken
2. Ein neues Passwort festlegen
3. Das neue Passwort zur bestätigung wiederholen
4. SSH key anzeigen mit folgendem Befehl:

**cat ~/.ssh/id\_rsa.pub**

1. SSH-Key in Zwischenablage kopieren
2. In GitHub einloggen
3. Auf das eigene Nutzerprofil gehen -> Einstellungen
4. *SSH und GPG keys* auswählen
5. Auf *Neuer SSH key* klicken
6. SHH-Key aus Zwischenablage einfügen und bestätigen

## Sensor- und Pumpen-/Ventileinheit

zum Flashen des Sketches auf einen Arduino muss dieser via USB-Kabel an einen PC angeschlossen werden. Wir verwenden zum Programmieren und auch zum Flashen die Arduino IDE (<https://www.arduino.cc/en/software>).

Für das Hydroponik-Projekt wurde ein Arduino Uno WiFi Rev2 beschafft. Folgende Schritte müssen in der Arduino IDE einmalig durchgeführt werden:

1. Werkzeuge > "Board: xyz" > Boardverwalter > **arduino megaAVR Boards** installieren
2. Werkzeuge > Bibliotheken verwalten > **ArduinoMqttClient** und **WiFiNINA** installieren
3. Werkzeuge > "Board: xyz" > Arduino megaAVR Boards > **Arduino Uno WiFi Rev2** auswählen
4. Optional: Ggf. muss unter Werkzeuge > Ports noch der korrekte COM-Port ausgewählt werden. Normalerweise passiert dies automatisch, wenn der Arduino angeschlossen wird.

Für die Nutzung von WLAN muss im Code der passende Zugang hinterlegt werden. Hierzu wird emfohlen, eine Datei “arduino\_secrets.h” anzulegen und dort die Felder zu definieren  
#define SECRET\_SSID DEIN\_WLAN\_NAME  
#define SECRET\_PASS DEIN\_WLAN\_PASSWORT

Ferner ist die IP-Adresse des MQTT Brokers (Raspberry Pi) zu hinterlegen. Dies geschieht aktuell im Code in Zeile 31 (Sensoreinheit) bzw. Zeile 19 (Pumpen\_Ventileinheit).