

Inhalt

| | |
|---|---|
| InfluxDB 2.x installieren und konfigurieren | 2 |
| Die Installation..... | 2 |
| Verwenden des CLI | 3 |
| Token in InfluxDB verwalten | 4 |
| DBRP Mapping..... | 4 |
| Einrichtungen in ioBroker | 5 |
| Grafana installieren | 6 |
| Zugriff aus Grafana auf InfluxDB | 6 |
| Grafana.ini Datei anpassen..... | 8 |

InfluxDB 2.x installieren und konfigurieren

Die offizielle Anleitung <https://docs.influxdata.com/influxdb/v2.3/install/?t=Raspberry+Pi> ist eher etwas für Eingeweihte, einfacher geht es mit dieser Anleitung <https://randomnerdtutorials.com/install-influxdb-2-raspberry-pi/>, welche auch direkt das CLI Werkzeug mitinstalliert.

Gleich drei Neuerungen machten mir das Leben schwer. Diese haben ihre Ursache in der Kommunikation mit *InfluxDB*: Die Steuerung von *InfluxDB* erfolgt ausschließlich über das API, das via http zugänglich ist.

- Der Befehl *Influx* öffnet keine spezifische Eingabeumgebung mehr. In der Version 2.x ist *influx* nun ein Befehl, der mit entsprechenden Kommandos, Optionen und Parametern versehen, die gewünschten Arbeiten ausführt.
- Die Authentifizierung erfolgt nicht mehr mittels Benutzer/Passwort Eingaben, sondern über sog. Token, die an das API gesendet werden müssen.
- Mit 2.x ist die Abfragesprache *InfluxQL* nicht mehr der Standard. Man soll dazu gedrängt werden künftig *Flux* zu verwenden.

Die Installation

Zum Installieren von *InfluxDB* genügen zwei Befehle im Terminal.

1.

```
wget -qO- https://repos.influxdata.com/influxdb.key | gpg --dearmor | sudo tee /etc/apt/trusted.gpg.d/influxdb.gpg > /dev/null export DISTRIB_ID=$(lsb_release -si); export DISTRIB_CODENAME=$(lsb_release -sc) echo "deb [signed-by=/etc/apt/trusted.gpg.d/influxdb.gpg] https://repos.influxdata.com/${DISTRIB_ID,,} ${DISTRIB_CODENAME} stable" | sudo tee /etc/apt/sources.list.d/influxdb.list > /dev/null
```



```
sudo apt-get update && sudo apt-get install influxdb2
```

Dass *InfluxDB* beim Booten ebenfalls als Service gestartet wird, bewirkt folgender Befehl ..

2.

```
sudo service influxdb start
```

Anschließend startet man die Weboberfläche (GUI) von *InfluxDB* über <http://<ip-Adresse>:8086>. Es startet ein Assistent, in dem man den künftigen Besitzer *admin/Passwort*, eine erste Organisation *db_iobroker* (Die *Buckets* werden in Organisationen verwaltet.) und eine erste Datenbank (*bucket*) anlegen sollte.)¹

Welcome — Initial User Setup — Complete

Setup Initial User

You will be able to create additional Users, Buckets and Organizations later

Username

Password Confirm Password

Initial Organization Name
An organization is a workspace for a group of users.

Initial Bucket Name
A bucket is where your time series data is stored with a retention policy.

[Continue](#)

¹ Username = admin; Password = !Creative17!; Initial Organization Name = org_galegro; Bucketname = db_iobroker

Verwenden des CLI

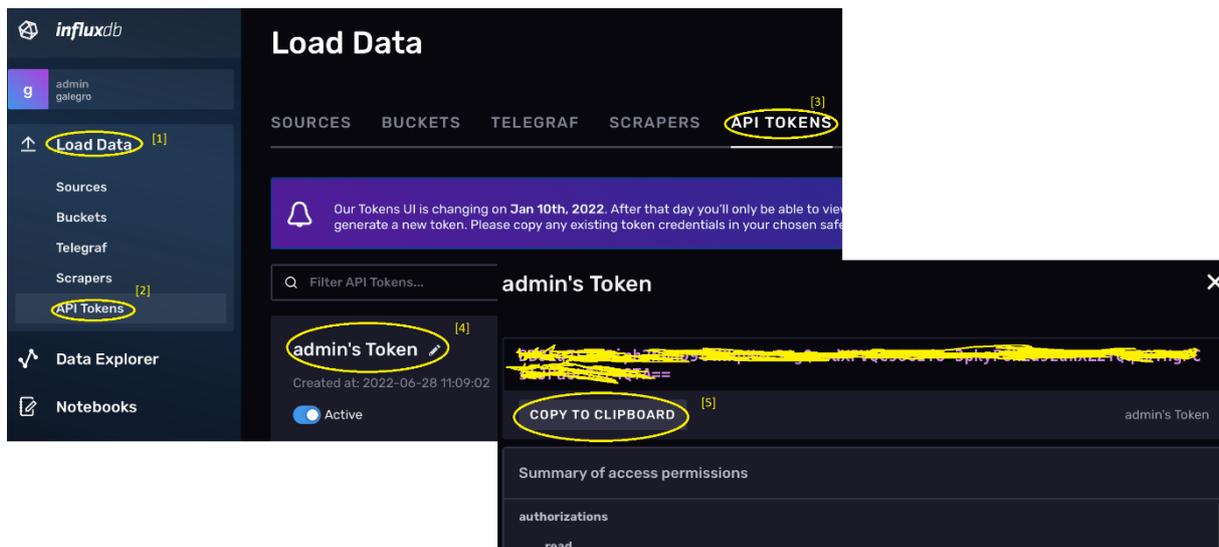
Da in *InfluxDB 2.x* Zugriffe auf die Daten ausschließlich über das API mittels Token erfolgen, ist eigentlich kein spezieller Benutzer für *ioBroker* mehr erforderlich. Wer dennoch einen solchen einrichten möchte, kann das über das CLI (Command Line Interface) bewerkstelligen. Verwendet man die originale Installationsanleitung, muss das CLI separat installiert werden.

Jeder Zugriff aus dem CLI heraus, muss mittels eines Token authentifiziert werden. Zur Vereinfachung dieses Vorganges sollte man sich eine Konfigurationsdatei anlegen, in der die nötigen Daten vorgehalten werden. Man kann durchaus mehrere solcher Dateien anlegen, die dann bei einem Aufruf mittels des Parameters `-t <Dateiname>` anzufügen sind. Wenn man eine solche Datei als *active* konfiguriert, werden die darin enthaltenen Informationen automatisch bei jedem *influx* Befehl hinzugefügt.

Das folgende Beispiel zeigt, wie man im Terminal eine solche Datei anlegen kann.)²

```
influx config create --config-name <NameDerConfigDatei> \  
--host-url http://localhost:8086 \  
--org <NameDerOrganisation> \  
--token <API-Token> \  
--active
```

Die hier aufgezeigte Vorgehensweise zeigt den Paradigmenwechsel bei Authentifizierungen weg von *user/password* und hin zur Verwendung von *Token*. Ist man wie zuvor beschrieben vorgegangen, existiert bloß das Token, welches beim Anlegen des Eigentümers *admin* der Datenbank automatisch angelegt wurde. Dies findet man wie in der Abbildung aufgezeigt.



```
influx config create --config-name galegro_admin_config \  
--host-url http://localhost:8086 \  
--org galegro \  
--token <Hier steht das aus der Zwischenablage reinkopierte Token> \  
--active
```

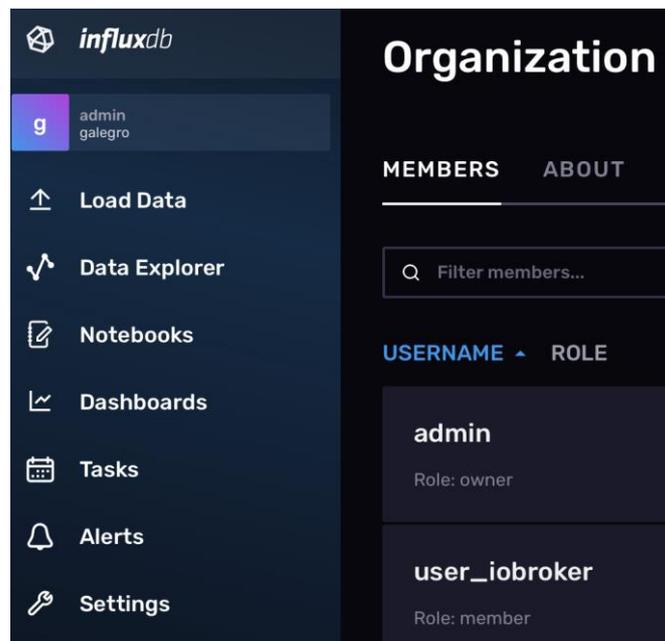
Die Konfigurationsdatei heißt *galegro_admin_config*, *InfluxDB* befindet sich auf dem lokalen Computer, mit dem man über das Terminal verbunden ist, die Organisation heißt *org_galegro* und die Datei wird als *active* konfiguriert.

Mit all diesen Daten kann man nun etwa einen Benutzer *user_iobroker* anlegen.

```
influx user create -n user_iobroker -p 'MeinPW123' -o galegro
```

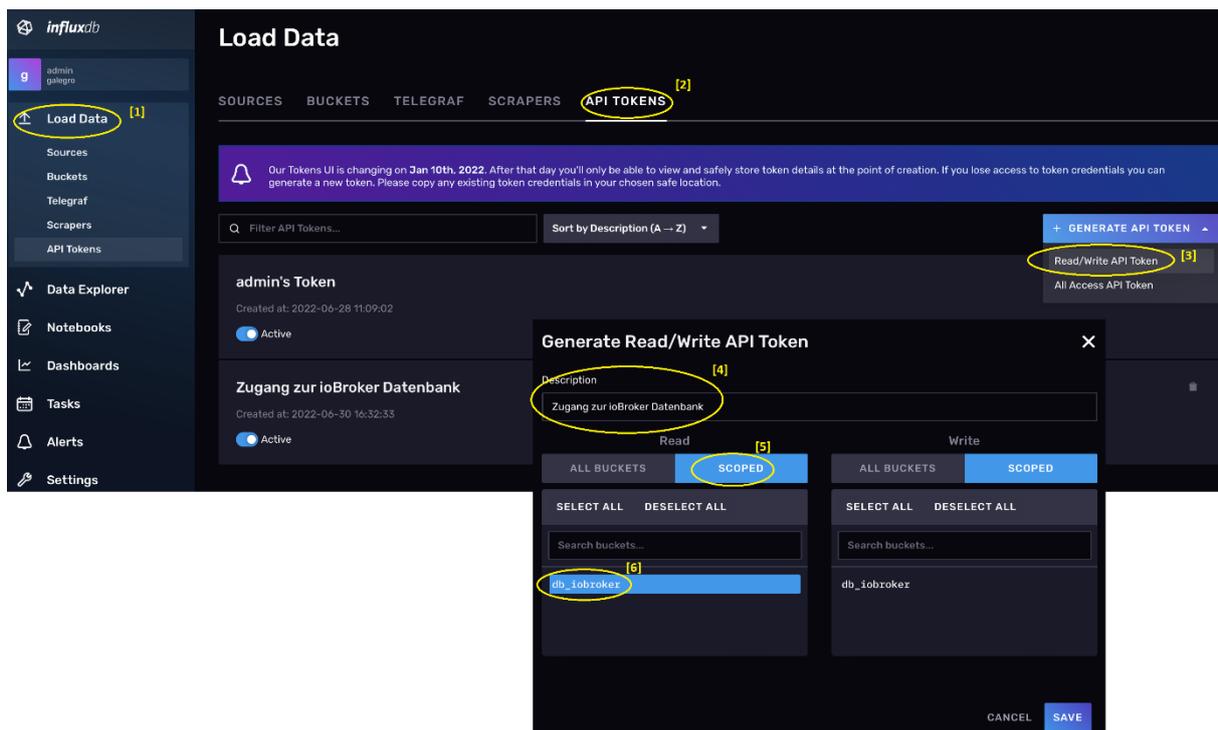
² Sollten die mittels „\“ getrennten Zeilen nicht fehlerfrei ausgeführt werden, muss alles in einer Zeile eingegeben werden; d.h.: Die „\“-Zeichen und Zeilenumbrüche müssen entfernt werden.

Das Ergebnis ist anschließend im GUI bzw. der Weboberfläche von *InfluxDB* zu bewundern.



Token in InfluxDB verwalten

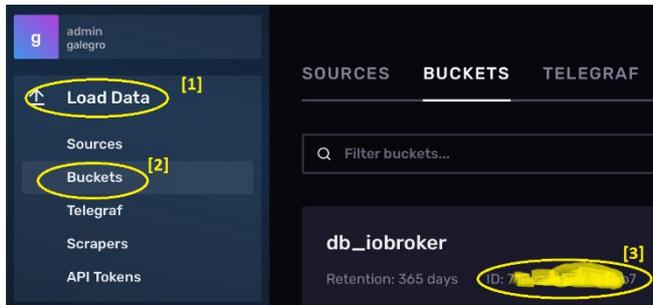
Da die Verwaltung der Datenbanken (Buckets) innerhalb von sog. Organisationen erfolgt, ist es sinnvoll, ein an eine solche Organisation gebundenes Token zu erzeugen. Die nachfolgende Abbildung zeigt das Vorgehen.



Nach den vorgenommenen Eintragungen ist dieses Token ausschließlich für den Lesezugriff auf die Datenbank *db_iobroker* ausgelegt.

DBRP Mapping

Möchte man mit der Abfragesprache *InfluxQL* von Grafana aus auf *InfluxDB* zugreifen, müssen die Datenbanken (*buckets*) einem sog. DBRP Mapping (**D**ata**B**ase and **R**etention **M**apping) unterzogen werden. Der Befehl ist nachfolgend aufgeführt. Für die obligatorische Angabe der sog. *Retention Policy* wird hier die bei der Installation von *InfluxDB* standardmäßig erzeugte Datei *autogen* verwendet.



```
influx v1 dbrp create \
--bucket-id 710eaaaa508256b7 \
--db db_jobroker \
--rp autogen \
--default
```

Einrichtungen in ioBroker



Zur Anbindung von *ioBroker* an eine *InfluxDB* Datenbank wird der Adapter *Influxdb* benötigt.

Der Adapter muss wie nachfolgend beschrieben konfiguriert werden. Als Voraussetzungen kommen hier die beim Installieren von *InfluxDB* getätigten Konfigurationen zum Tragen.

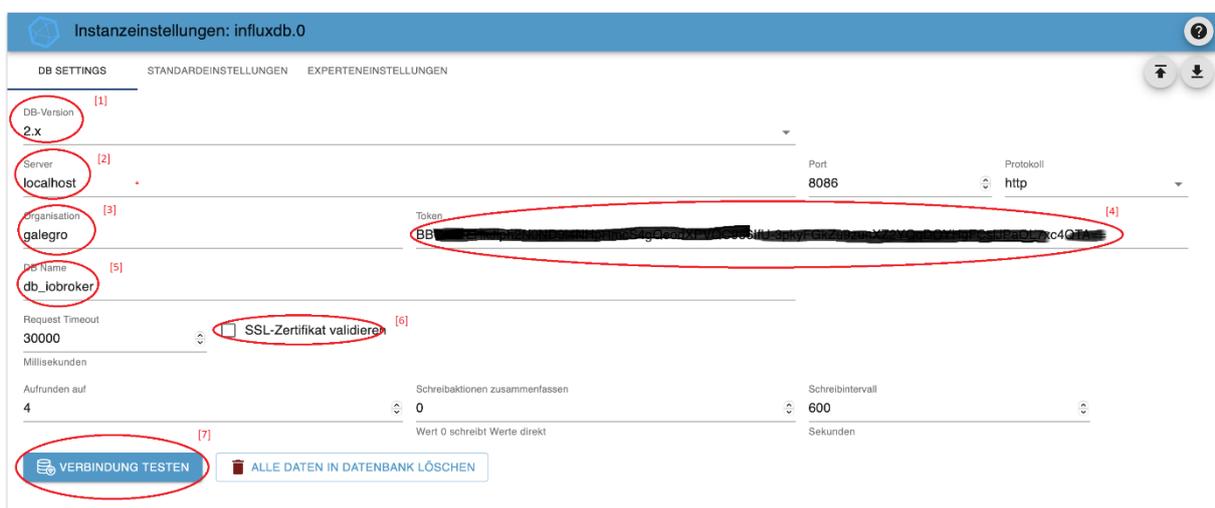
Token zur Organisation *org_galegro* oder auch jenes vom Datenbankbesitzer *admin* ist möglich

Organisation: *galegro*

Zugriff: http, Port 8086

Bucket: Es wurde ein Bucket namens *db_jobroker* eingerichtet.

1. Die Datenbank ist von der Version 2.x
2. Die Datenbank befindet sich auf demselben Raspberry Pi wie ioBroker. Die Einstellungen von Port und http müssen wie gezeigt übernommen werden.
3. Bei der Installation musste eine Organisation und ein Besitzer eingerichtet werden. Hier ist der Name der Organisation einzugeben.
4. Hier wird ein zuvor eingerichteter Token eingetragen, das unbeschränkten Zugriff erlaubt. Das Admin-Token sollte hier stehen. Ein Bucket-Token reicht hier nicht aus.
5. Hier wird die Datenbank (Bucket) eingetragen, in die ioBroker schreiben soll.
6. Da als Zugriffsprotokoll http gewählt wurde, ist SSL zu deaktivieren.
7. Hat man alles richtig eingetragen, so sollte ein Test erfolgen, ob auf die Datenbank zugegriffen werden kann.



Grafana installieren

Hier <https://grafana.com/tutorials/install-grafana-on-raspberry-pi/> ist die original Anleitung zu finden.

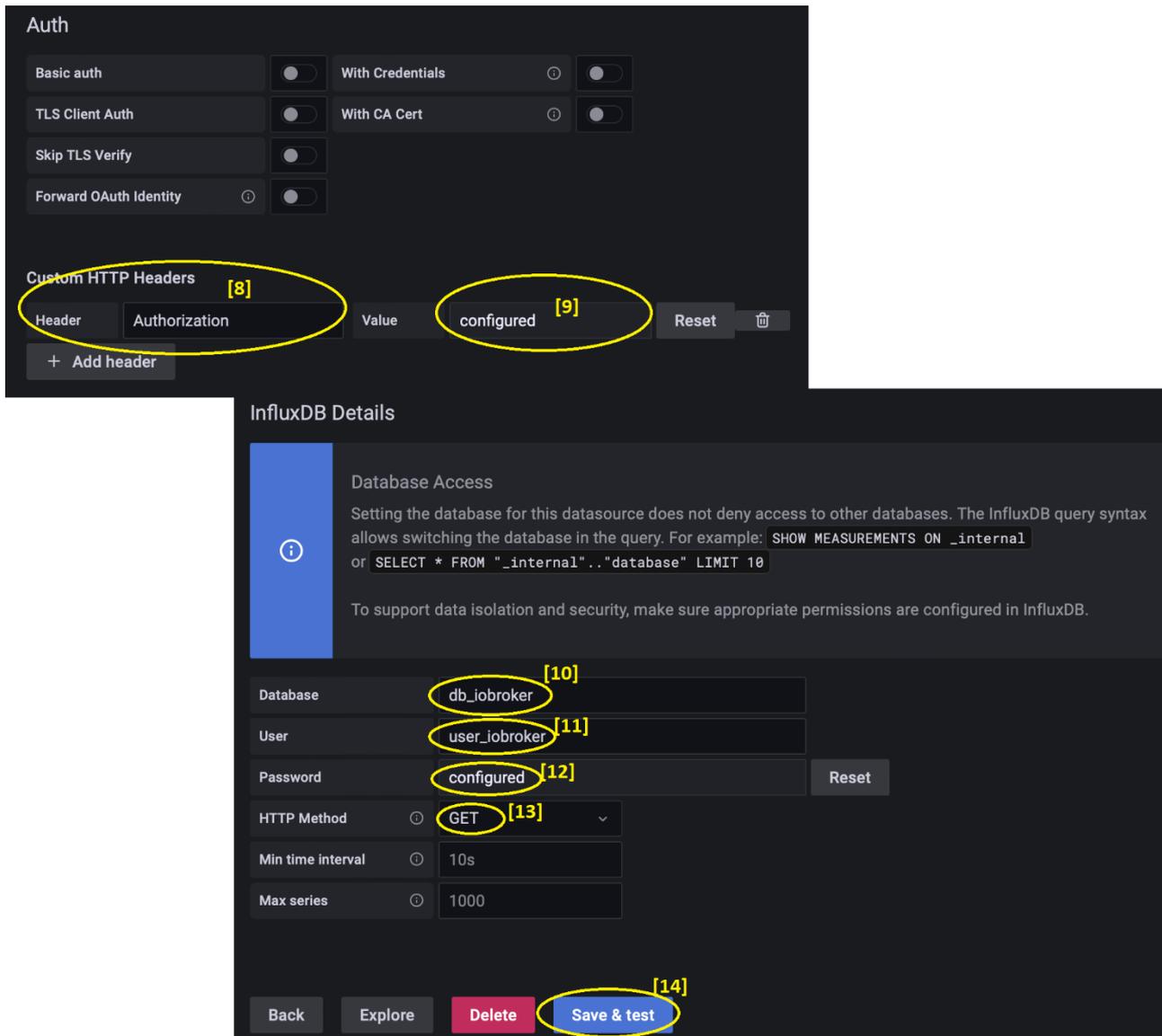
1. Installieren des APT Schlüssels mittels ..
`wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -`
2. Hinzufügen des Grafana APT Repository ..
`echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee -a /etc/apt/sources.list`
3. Hinzufügen des APT Repositorys ..
`echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee -a /etc/apt/sources.list`
4. Grafana installieren ..
`sudo apt-get update`
`sudo apt-get upgrade`
5. Nun ist Grafana installiert. Starten von Grafana beim Booten ..
`sudo /bin/systemctl enable grafana-server`
`sudo /bin/systemctl start grafana-server`

Nun kann man sich in Grafana mittels user/password = admin/admin unter <http://<Host-IP>:3000> einloggen.

Zugriff aus Grafana auf InfluxDB

The image shows two screenshots from the Grafana web interface. The top screenshot shows the 'Configuration' page with the 'Data sources' tab selected. A yellow circle highlights the 'Add data source' button, labeled with a '[2]'. Another yellow circle highlights the gear icon in the left sidebar, labeled with a '[1]'. The bottom screenshot shows the 'Add data source' dialog box with 'InfluxDB' selected. A yellow circle highlights the 'InfluxDB' option, labeled with a '[3]'. Below this, the 'Data Sources / InfluxDB' configuration page is shown. A yellow circle highlights the 'Name' field set to 'InfluxDB', labeled with a '[4]'. Another yellow circle highlights the 'Query Language' dropdown set to 'InfluxQL', labeled with a '[5]'. A third yellow circle highlights the 'URL' field set to 'http://192.168.4.58:8086', labeled with a '[6]'. A fourth yellow circle highlights the 'Access' dropdown set to 'Server (default)', labeled with a '[7]'. The 'Auth' section at the bottom is also visible.

Die Einstellungen in *Grafana* sind in obiger Abbildung dargestellt. In [1] wird die Option *data source* selektiert. Anschließend wählt man über die Schaltfläche *Add data source* [2] aus den angebotenen Optionen den Eintrag *InfluxDB* [3] aus. In dem sich öffnenden Dialog trägt man in [4] die frei wählbare Bezeichnung für die Datenbank ein. Als *Query Language* ist *InfluxQL* zu wählen [5]. In [6] ist die URL zu InfluxDB (UI) einzutragen. Als Zugriff wird üblicherweise *Server* gewählt [7].



Damit der Zugriff auf *InfluxDB* möglich wird, sind in [8] der Befehl *Authorization* und in [9] ein Token ⁴ einzutragen, das den Zugriff auf die *Organisation* gewährt. Als Token sollte man geeignetes, zuvor erstelltes, spezielles Zugriffstoken erstellt haben. (siehe: [Token in InfluxDB verwalten](#) auf Seite 3)

Leider ist es damit noch nicht getan. Beim klassischen Zugriff auf eine Datenbank in *InfluxDB* muss nach wie vor der Datenbankname sowie ein Benutzername und Passwort verwendet werden. Kann die Datenbank als sog. Bucket im UI ⁵ von *InfluxDB* erstellt werden, so muss ein solcher weiterer [Verwenden des CLI](#) ⁶ eingerichtet werden. Diese Daten sind in [10] bis [12] einzutragen. Als Zugriff muss GET gewählt werden.

Ein Klick auf *Save & test* sollte hoffentlich eine erfolgreiche Anbindung an *InfluxDB* signalisieren.

Ein Klick auf *Save & Test* [1] sollte anschließend den erfolgreichen Verbindungsaufbau [2] zu *InfluxDB* aufzeigen.

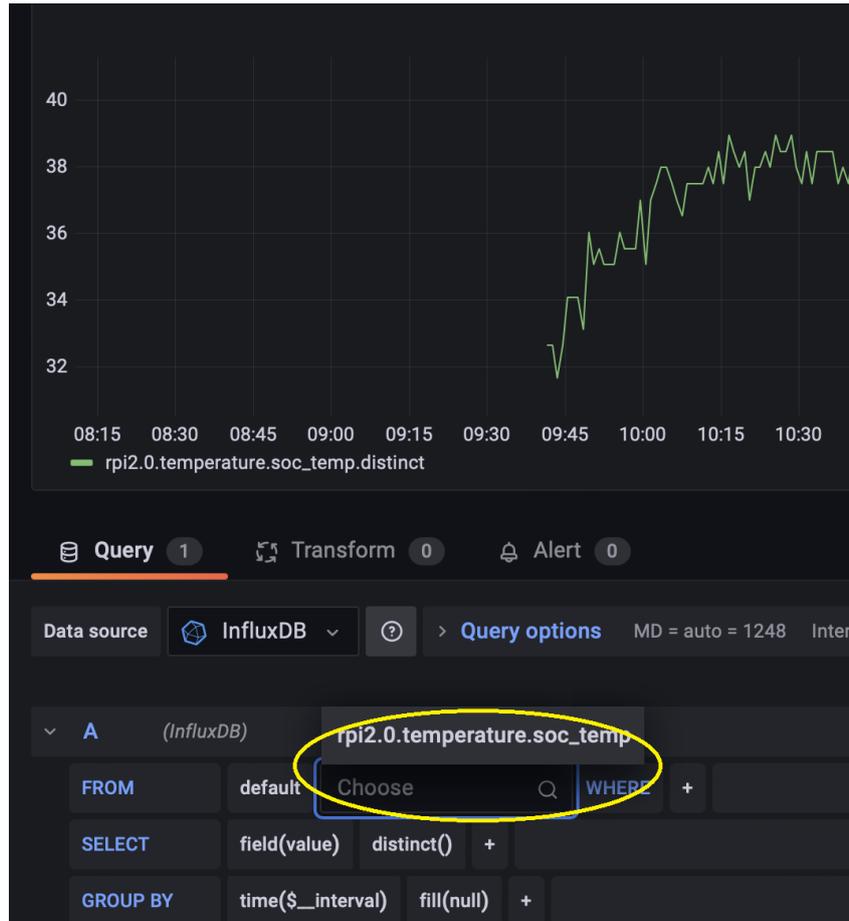
³ Dies hat zur Folge, dass die Datenbank (Bucket) einem sog. *DBRP Mapping* unterzogen werden muss. *InfluxDB 2.x* verwendet standardmäßig die für die Abfragesprache *Flux* konfigurierten Buckets.

⁴ Schlüsselwort *Token* gefolgt vom eigentlich Token. Beispiel: *Token AghK34..KrPPl ...*

⁵ User Interface -> Weboberfläche

⁶ Command Line Interface

Trotz erfolgreicher Verbindung werden bei der Erstellung eines (neuen) Dashboards in *Grafana* unter *measurements* (siehe FROM clause) keine Datenreihen angezeigt, wenn man nicht zuvor das *MBRP Mapping* für die Datenbank (Bucket) durchgeführt hat, welche diese Datenreihen enthält.



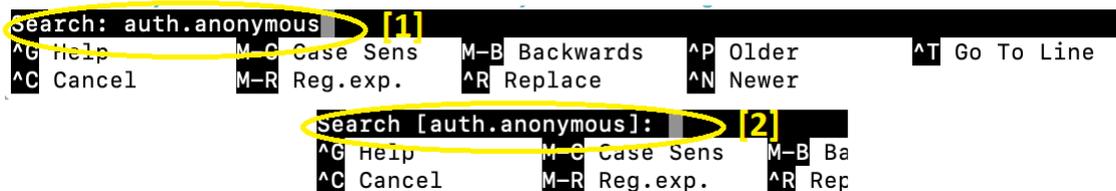
Grafana.ini Datei anpassen

Damit die mit Grafana erstellten Darstellungen auch in der Visualisierung der VIS von ioBroker eingebettet werden können, ist eine Bearbeitung der Datei *grafana.ini* erforderlich. Ihre Bearbeitung kann beispielsweise mit dem Editor *nano* erfolgen.

```
sudo nano /etc/grafana/grafana.ini
```

Insgesamt müssen in drei Abschnitten *security*, *auth.anonymous* und *users* Änderungen erfolgen. Hierbei ist zu beachten, dass das Semikolon am Anfang der Zeile entfernt wird, ansonsten wird die Zeile als Kommentar angesehen und bleibt unberücksichtigt.

Durch Drücken von **Ctrl+W** und Eingabe des Suchbegriffs [1], kann man schnell zu der gesuchten Stelle finden. Ist die gefundene Stelle nicht die richtige, so kann die Suche durch erneutes Drücken von **Ctrl+W** fortgesetzt werden. In diesem Fall schlägt *nano* den zuletzt verwendeten Suchbegriff vor; durch Drücken auf die Eingabetaste, wird die nächste Stelle gesucht, an der der Begriff zu finden ist.



Die erste Änderung im Bereich *security* ermöglicht das Einbetten von Elementen aus *Grafana* in andere Anwendungen.

```
[security]
allow_embedding = true
```

Im Bereich *auth.anonymous* wird geregelt, wie auf ein eingebettetes Objekt aus Grafana in einer anderen Anwendung zugegriffen werden darf. Die erste Option erlaubt ein Zugriff, ohne dass sich der Anwender authentifizieren muss. Alle Objekte werden in Grafana in sog. Organisations vorgehalten. Hier ist jene einzugeben, unter der das Objekt in Grafana geführt wird. Die dritte Option weist dem Anwender die Zugriffsrechte zu; im hier dargestellten Beispiel kann der Anwender keine Änderungen am übergebenen Objekt vornehmen.

```
[auth.anonymous]
enabled = true
org_name = org_iobroker
org_role = Viewer
```

Um beim Zugriff auf ein eingebettetes Objekt sich nicht gegenüber *Grafana* authentifizieren zu müssen, sollte man im Bereich *users* dies ausschalten.

```
[users]
allow_sign_up = false
```

Damit nun alle vorgenommenen Einstellungen auch wirksam werden, muss Grafana neu gestartet werden. Dies kann man auf zweifache Weise erreichen. Entweder ..

```
sudo /bin/systemctl restart grafana-server
```

.. oder ..

```
sudo service grafana-server stop
sudo service grafana-server start
```