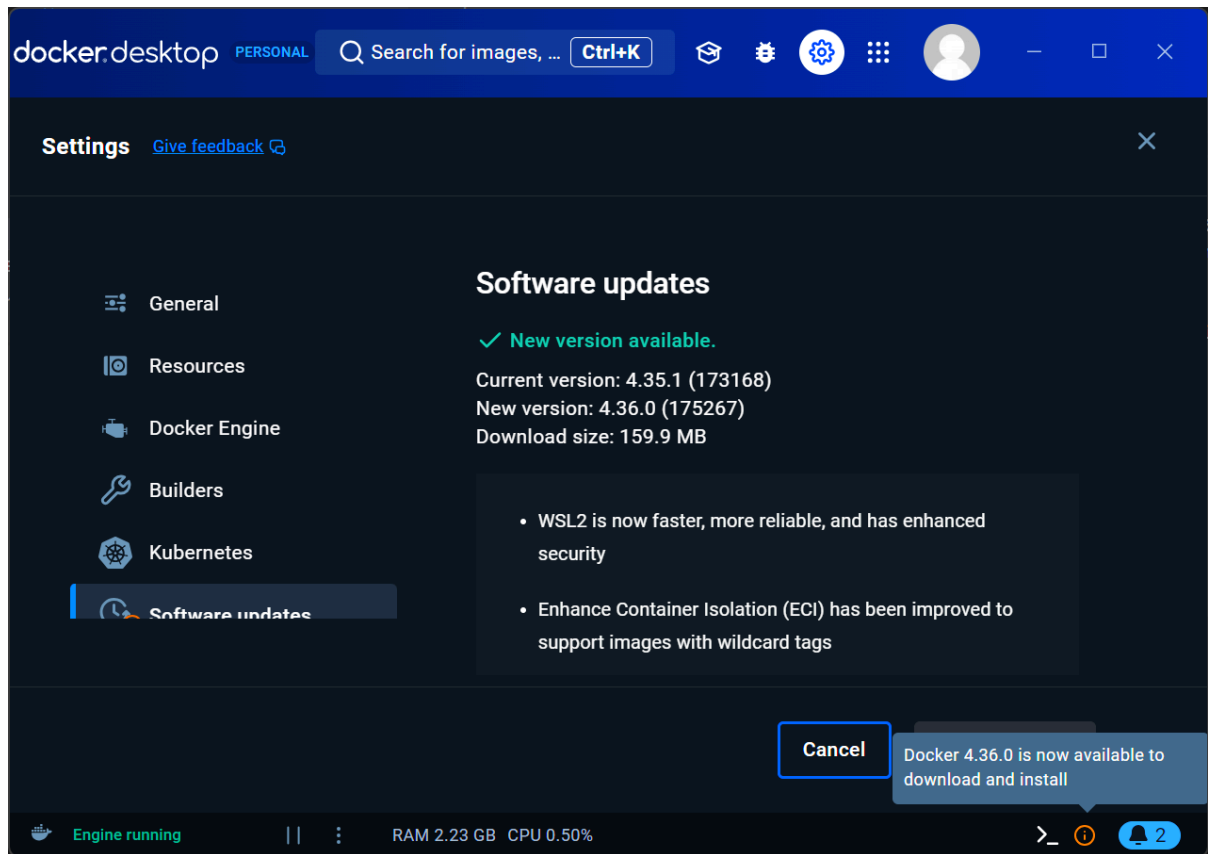


# Installation von InfluxDB mit docker

Führen Sie folgende Schritte aus:

## Start Docker-Desktop auf Notebook

Stellen Sie sicher, dass Sie eingeloggt sind.



*Docker-Desktop aktualisiert und angemeldet*

Führen Sie ein update des Docker-Desktops aus, falls das notwendig ist.

## Verzeichniss erstellen für Ihre Docker-Files

```
mkdir docker-compose-files
cd .\docker-compose-files\
mkdir influxdb
cd influxdb
```

## Erstellung des `docker-compose.yml` -Files

```
docker-compose.yml
```

```

services:
  influxdb:
    image: influxdb:2
    container_name: influxdb
    restart: always
    ports:
      - '8086:8086'
    environment:
      DOCKER_INFLUXDB_INIT_MODE: setup
      DOCKER_INFLUXDB_INIT_USERNAME: admin
      DOCKER_INFLUXDB_INIT_PASSWORD: btw2201btw2201
      DOCKER_INFLUXDB_INIT_ORG: BFH
      DOCKER_INFLUXDB_INIT_BUCKET: OpenWeather
    volumes:
      - ./data:/var/lib/influxdb2
      - ./config:/etc/influxdb2

```

## Docker-Container starten

Im Verzeichnis, wo das `docker-compose.yml` liegt, starten Sie die Container mit

```
docker compose up -d
```

Kontrolle, ob Container läuft mit

```
docker ps -a
```

CONTAINER ID	IMAGE NAMES	COMMAND	CREATED	STATUS	PORTS	
4de7409b8226	influxdb:2	"/entrypoint.sh infl..."	26 hours ago	Up 3 minutes	0.0.0.0:8086->8086/tcp	influxdb
fc1caca58af8	postgres:latest	"docker-entrypoint.s..."	5 days ago	Up 3 minutes	0.0.0.0:5432->5432/tcp	postgresql-latest
6cd29c6d37a9	dpag/pgadmin4	"/entrypoint.sh"	5 days ago	Up 3 minutes	443/tcp, 0.0.0.0:5050->80/tcp	pgadmin4
2e33811a433f	phpmyadmin/phpmyadmin:latest	"/docker-entrypoint..."	6 weeks ago	Up 3 minutes	0.0.0.0:80->80/tcp	phpmyadmin-container
713f635b2113	mysql:8.0.40	"docker-entrypoint.s..."	6 weeks ago	Up 3 minutes	0.0.0.0:3306->3306/tcp, 33060/tcp	mysql

Container running?

Auch im Docker-Desktop sind die laufenden Container sichtbar:

**Containers** [Give feedback](#)

Container CPU usage 78.65% / 800% (8 CPUs available)      Container memory usage 726.7MB / 30.49GB      [Show charts](#)

Search  Only show running containers

	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	influxdb	-	-	-	0.14%	14 seconds ago	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	influxdb	4de7409b8226	influxdb:2	8086:8086	0.14%	14 seconds ago	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	postgresql	-	-	-	78.2%	14 seconds ago	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	postgresql-latest	fc1caca58af8	postgres:latest	5432:5432	0.02%	14 seconds ago	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	pgadmin4	6cd29c6d37a9	dpag/pgadmin4	5050:80	78.18%	14 seconds ago	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	mysql	-	-	-	0.3%	14 seconds ago	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

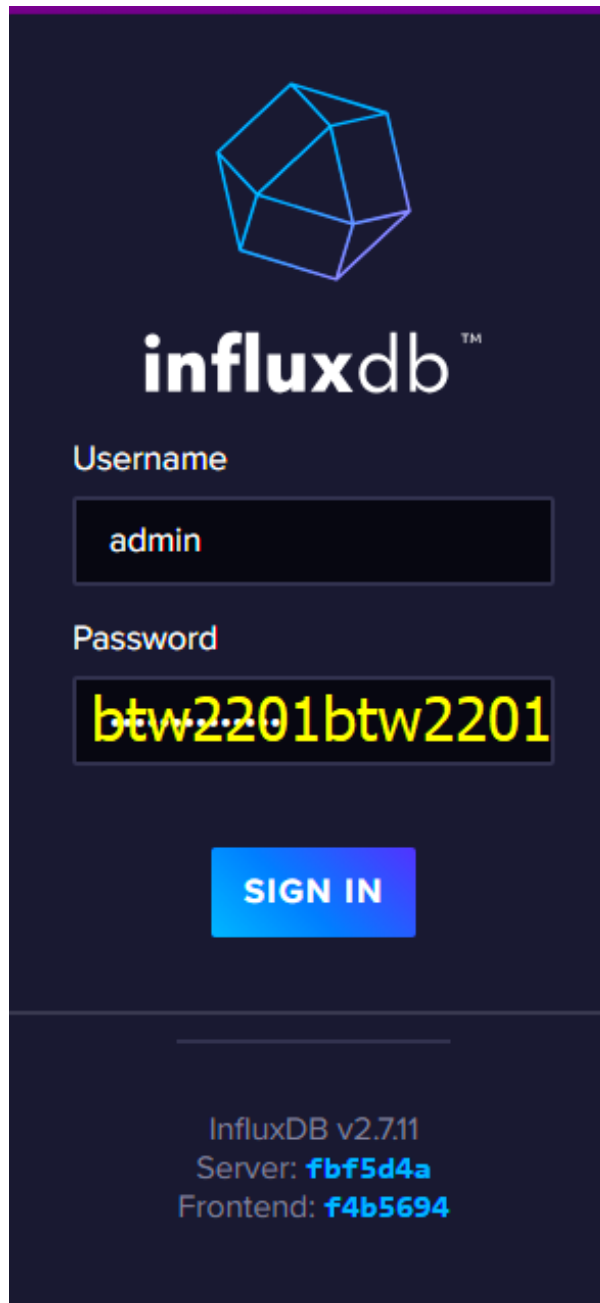
Showing 8 items

Container running?

## Verbindung mit GUI von InfluxDB

Das Web-GUI von InfluxDB erreichen wir nun auf `http://localhost:8086`

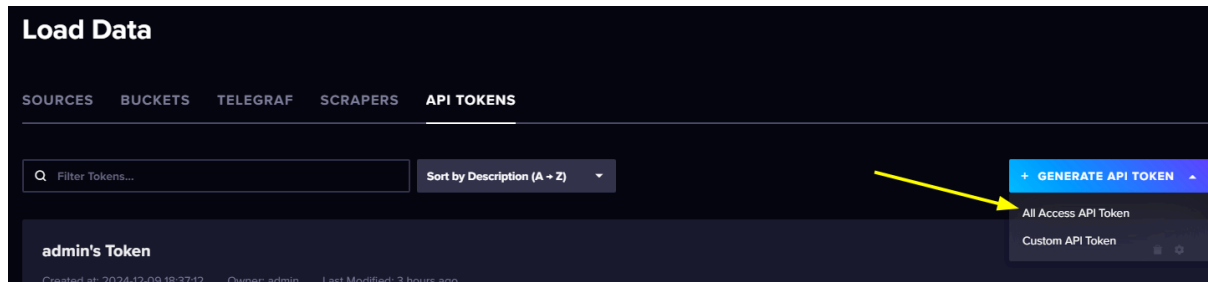
Login mit user `admin`, Passwort: `btw2201btw2201`. Diese Credentials haben wir im `docker-compose.yml` definiert.



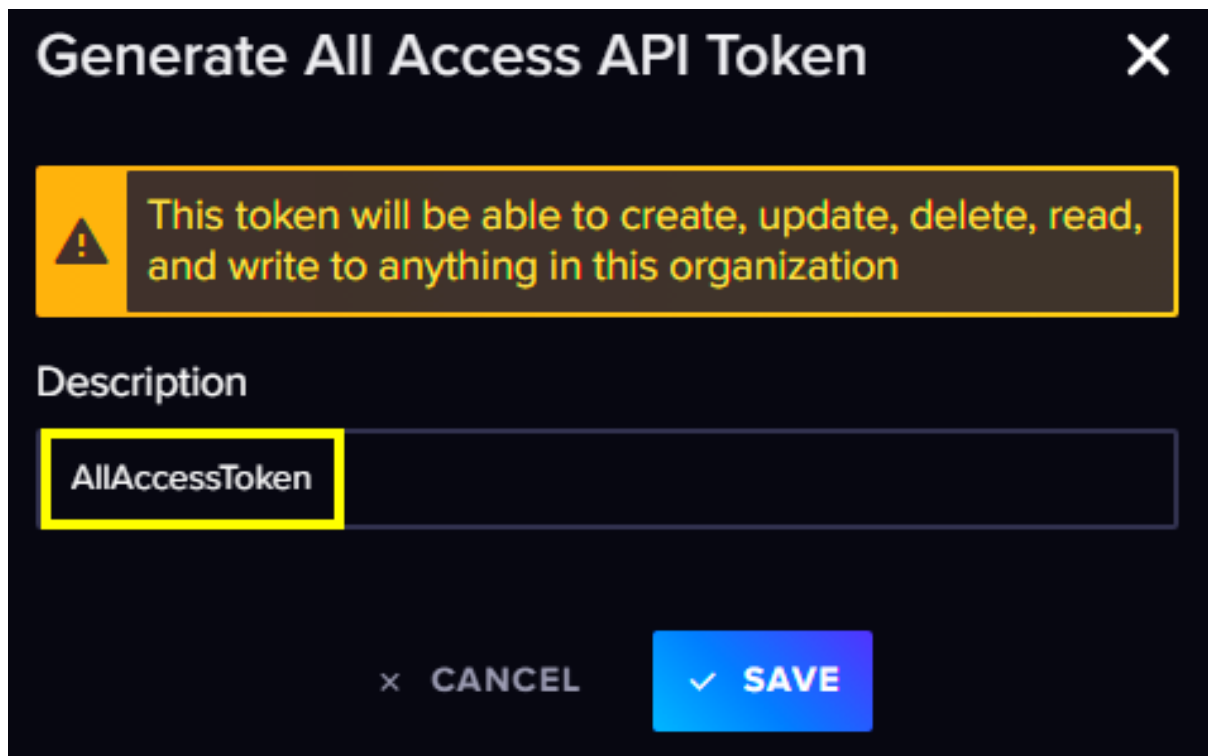
Login WebGUI InfluxDB

## API Token generieren

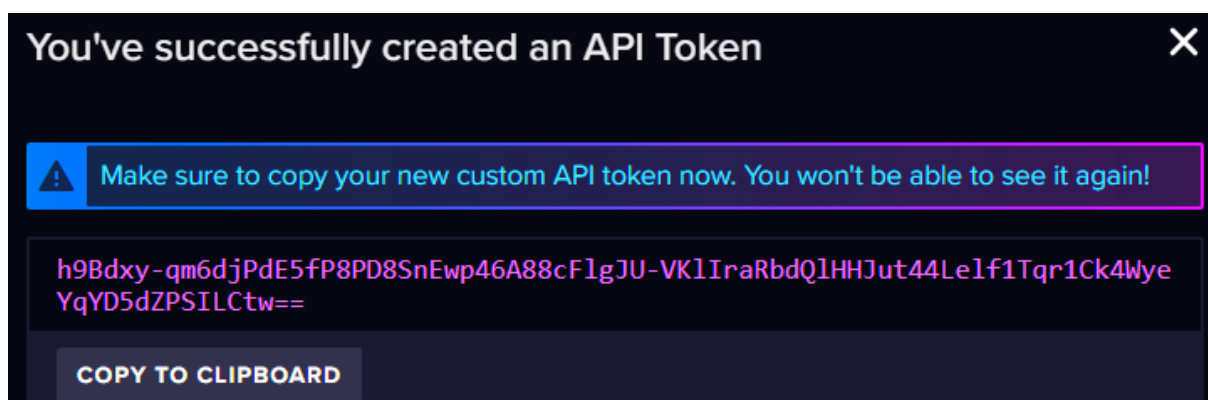
Generiere dir als Erstes ein API Token. Dieses werden wir für alles Weitere als Zugangskey verwenden.



All Access API Token generieren



Das bei Dir generierte Token muss notiert werden!

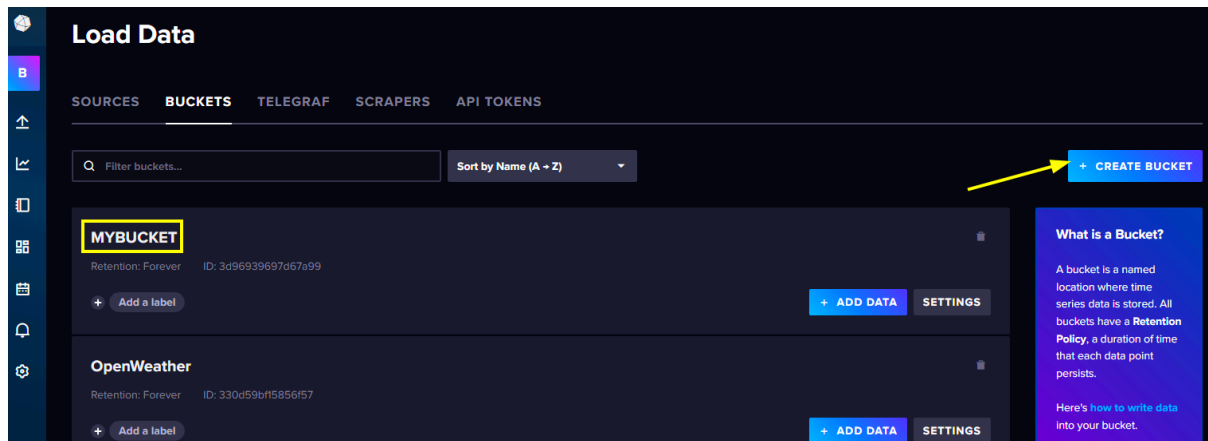


Beispieltoken ⚠ NICHT kopieren ist ein Beispiel! ⚠

```
h9Bdxy-qm6djPdE5fP8PD8SnEwp46A88cFlgJU-VK1IraRbdQlHHJut44Le1f1Tqr1Ck4WyeYqYD5dZPSILCtw==
```

Bucket zum Testen erstellen

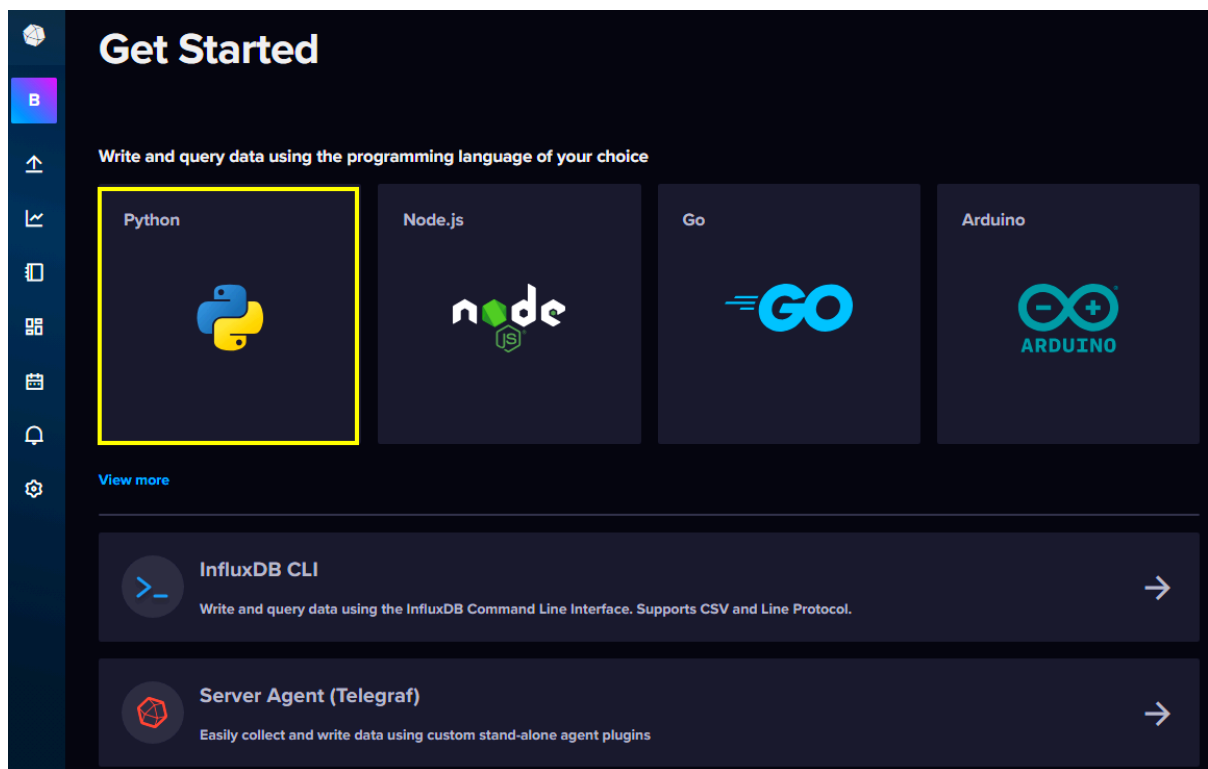
Erstellen Sie auch einen Testbucket, z.Bsp mit Namen `MYBUCKET`



*Bucket erstellen*

## Introbeispiel

Führen Sie das Python Programmierbeispiel durch. Verwende jedoch immer Dein generiertes Token von oben.



*Write and query data using Python*

#### Note

- Ihr **URL** lautet: `url="http://localhost:8086"`
- **Token:** `token="IHR TOKEN"`
- **Organisation:** `org="BFH"`
- **Bucket:** `bucket="MYBUCKET"`
- Initialisierungsstring client: `client = InfluxDBClient(url=url, token=token, org=org)`

Angelehnt an das Beispiel in *Get Started* würde ein Skript zum Schreiben der Testdaten so ausschauen

```
import influxdb_client, os, time
from influxdb_client import InfluxDBClient, Point, WritePrecision
from influxdb_client.client.write_api import SYNCHRONOUS

client = InfluxDBClient(url="http://localhost:8086", token="IHR TOKEN", org="BFH")

bucket="MYBUCKET"

write_api = client.write_api(write_options=SYNCHRONOUS)

for value in range(5):
    point = (
        Point("measurement1")
        .tag("tagname1", "tagvalue1")
        .field("field1", value)
    )
    write_api.write(bucket=bucket, org="BFH", record=point)
    time.sleep(1) # separate points by 1 second
```