



Zuyd Hogeschool
BD01

Fysiek handleiding

Groepsleden:
Dylan Giesberts
Rik Heijmann
Jordy Clignet

Opdrachtgever:
Maarten Vaessen

Begeleidende docent:
Marcel Schmitz

Document informatie

Versiebeheer

Vastleggen wanneer iemand aan het document heeft gewerkt en wat veranderd is.

| Versie | Gewijzigd | Auteur | Datum |
|--------|-------------------------------|--------|------------|
| 0.1 | Initiële template aangeleverd | Allen | 15-09-2021 |

Distributielijst

Wanneer is welke versie uitgegeven/ingeleverd en door wie.

| Versie | Omschrijving | Distributie | Datum |
|--------|---------------------------------------|-------------|-----------|
| 0.1 | Delen templatedocument aan casusgroep | Samen | 15-9-2021 |

1. Informatiepagina

Studenten: Dylan Giesberts, Rik Heijmann, Jordy Clignet.

Opleiding: Zuyd hogeschool, Bachelor ICT

Schoolbegeleider/procesbegeleider:

Periode: Blok 2 van studiejaar 2021-2022

Opdrachtgever: Zuyd Hogeschool en Politie Nederland

Datum en plaats: 15-9-2021, Heerlen

Inhoudsopgave

| | |
|---|-----------|
| 1. INFORMATIEPAGINA..... | 2 |
| 2. VOORWOORD..... | 3 |
| 3. SAMENVATTING..... | 4 |
| 4. AFKORTINGEN..... | 6 |
| 5. INLEIDING..... | 7 |
| 5.1. LEESWIJZER..... | 7 |
| 5.2. CONTEXT..... | 8 |
| 6. FUNCTIONEEL ONTWERP..... | 9 |
| 6.1. HET MEETKASTJE..... | 9 |
| 6.2. VAN MEETKASTJE TOT DATABASE..... | 10 |
| 6.1. VAN DATABASE TOT DATA SCIENTIST..... | 10 |
| 7. TECHNISCHE ONTWERP..... | 12 |
| 7.1. MEETKASTJE..... | 12 |
| 7.1.1. Elektrischschema..... | 12 |
| 7.1.2. Microcontroller firmware..... | 12 |
| 7.2. MQTT BROKER..... | 14 |
| 7.2.1. Docker..... | 14 |
| 7.2.1. Firewall regels..... | 14 |
| 7.1. DATABASE..... | 15 |
| 7.1.1. Docker..... | 15 |
| 7.1.1. Firewall regels..... | 17 |

2. Inleiding

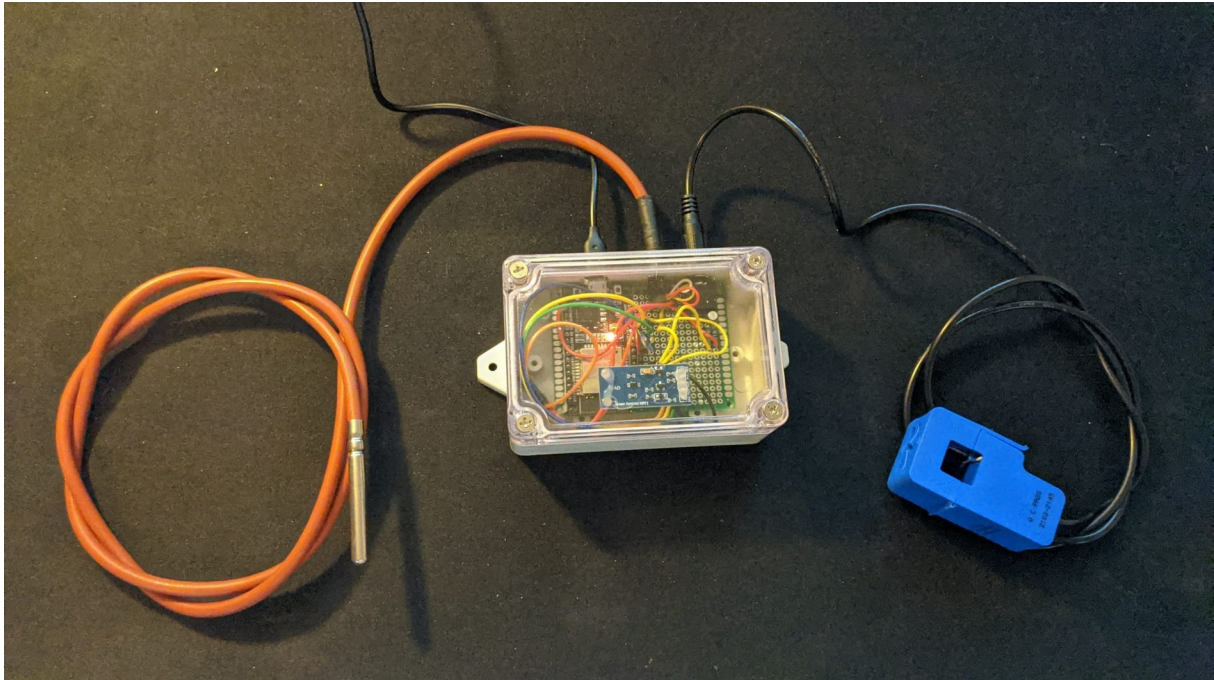
Deze set aan instructies helpt u met het configureren en plaatsen van uw zonnepaneelmeetkastje. Daarnaast beschrijft dit document ook de stappen om uw eigen servers uit te voeren.

3. Afkortingen

| Afkorting | Uitleg |
|-----------|---|
| IP-adres | Internet Protocol adres, een IPv4- of IPv6-adres afhankelijk van uw situatie. |

Tabel 1: Lijst met afkortingen

4. Instructies

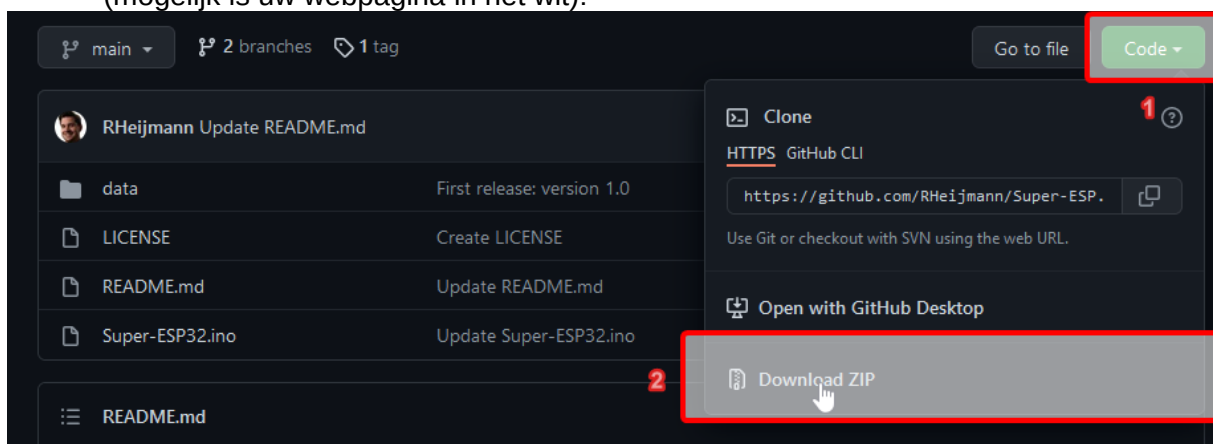


Figuur 1: Temperatuursensor links (bruine kabel met metalen punt), meetkastje met lichtsensor in het midden (lichtsensor zit op de blauwprintplaat) en de stroomsensor aan de rechterkant.

4.1. Het installeren van het meetkastje

4.1.1. Indien u nog geen firmware heeft

1. Indien u nog niet de juiste firmware hebt ontvangen en deze dus ook nog niet op uw kastje is geïnstalleerd of u wilt uw configuratie aanpassen dan navigeert u naar: <https://github.com/RHeijmann/Super-ESP/>
Download hierbij de gehele GitHub-folder. Dit doet u door het volgende uit te voeren (mogelijk is uw webpagina in het wit):



Figuur 2: Hoe download u een GitHub folder?

2. Extraheer het gedownloade zip-bestand.

3. Indien nog niet geïnstalleerd installeer: <https://www.arduino.cc/en/software>

4. Open vanuit de geëxtraheerde zip-bestand middels Arduino het bestand: Examples/Solarpanel/Super-ESP32.ino

5. Ben ervan verzekerd dat de volgens boards zijn geïnstalleerd. Dit kunt u vanuit Arduino controleren door in de menubalk op "Tools" > "Board:...". Indien "Board:..." gelijk staat aan "Board: DOIT ESP32 DEVKIT V1" heeft u de juiste firmware geïnstalleerd. Zo niet dat klikt u op "File" > "Preferences". Hierin vult u bij het stukje "Additional Boards Manager URLs" het volgende in: https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_dev_index.json. Daarna klikt u op: "Tools" > "Board:..." > "Boards Manager". In het zoekvenster vult u "esp32" in. Installeer hier de 2.0.2 versie van "Espressif systems". Als laatste klikt u op "Tools" > "Board:..." > "ESP32 Arduino" > "DOIT ESP32 DEVKIT V1". Indien "Board:..." Nu heeft u de juiste firmware geïnstalleerd.

6. Ben ervan verzekerd dat de volgende libraries zijn geïnstalleerd. Dit doet u door op "Tools" > "Manage Libraries" te klikken. Zodra u op de naam van een van deze libraries zoekt krijgt u ook direct de optie om deze te installeren.

| # | Naam | Versie | Waarvoor wordt het gebruikt? | Auteur | Bron |
|---|--------------|--------|---|----------------------|-----------------------------|
| 1 | ArduinoJson | 6.18.5 | Het opslaan en openen van de vaste configuratie. | Benoit Blanchon | Arduino IDE Library Manager |
| 2 | BH1750 | 1.2.0 | Het meten van licht, middels de BH1750 sensor. | Christopher Laws | Arduino IDE Library Manager |
| 3 | DS18B20 | 1.0 | Het meten van temperatuur, middels de DS18B20 sensor. | Matthias Munk Hansen | Arduino IDE Library Manager |
| 4 | EmonLib | 1.1.0 | Het meten van stroom, middels de SCT013-050 sensor. | OpenEnergyMonitor | Arduino IDE Library Manager |
| 5 | OneWire | 2.3.6 | Ter ondersteuning van DallasTemperature library. | Paul Stoffregen | Arduino IDE Library Manager |
| 6 | PubSubClient | 2.8.0 | Voor het verzenden van MQTT-berichten | Nick O'Leary | Arduino IDE Library Manager |

7. Ben ervan verzekerd dat deze Arduino libraries zijn geplaatst in de folder: "C:\Users\[Uw Gebruiker]\Documents\Arduino\libraries". Indien niet aanwezig dient u deze ook te downloaden vanuit GitHub.

| # | Naam | Versie | Waarvoor wordt het gebruikt? | Auteur | Bron |
|---|-------------|--------|--|-------------------|--|
| 1 | ESP8266Ping | 1.0 | Voor het testen of dat er een internetverbinding is op | Daniele Colonardi | GitHub: https://github.com/danco190/ESP8266Ping |

| | | | | | |
|---|-----------|-----|--|--------------------|---|
| | | | ES8266 gebaseerde apparaten. | | |
| 2 | ESP32Ping | 1.6 | Voor het testen of dat er een internetverbinding is op ESP32 gebaseerde apparaten. | Marian Craciunescu | GitHub: https://github.com/marian-craciunescu/ESP32Ping |

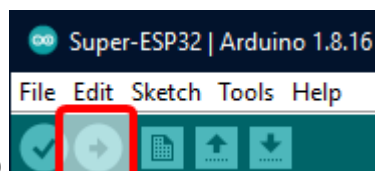
8. Zoek en vervang de volgende waarden om uw eigen firmware in te stellen:

| Zoek | Vervang met |
|---|---|
| #define OWNERID 1 | #define OWNERID [Uw owner ID] |
| #define DEVICEID 1 | #define DEVICEID [Uw device ID] |
| const String installationDate = "2022-01-20 15:20:00" | const String installationDate = "[Huidige datum en tijdstip in het formaat: JJJJ-MM-DD UU:mm:ss]" |

9. Open het "Examples/Solarpanel/data/configuration.json"-bestand in kladblok of een soort gelijke tekstbewerker.

10. Zoek en vervang de volgende waarden om uw apparaat te configureren:

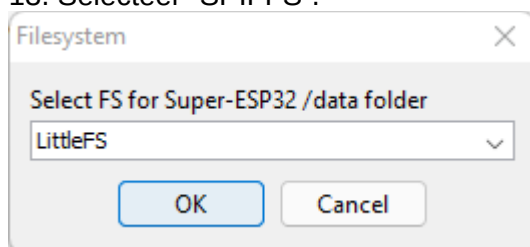
| Zoek | Vervang met |
|--------------------------------|---|
| "hostname": "Solarpanel0001", | "hostname": "Solarpanel[Uw device ID]", |
| "ssid": "WiFinaam", | "ssid": "[Het SSID van uw WiFi-netwerk]", |
| "password": "WiFi-wachtwoord", | "password": "[Het wachtwoord van uw WiFi-netwerk]", |



11. Klik op . Nu wordt de firmware op uw meetkastje geïnstalleerd.

12. Klik op "Tools" > "ESP32 Sketch Data Upload".

13. Selecteer "SPIFFS".

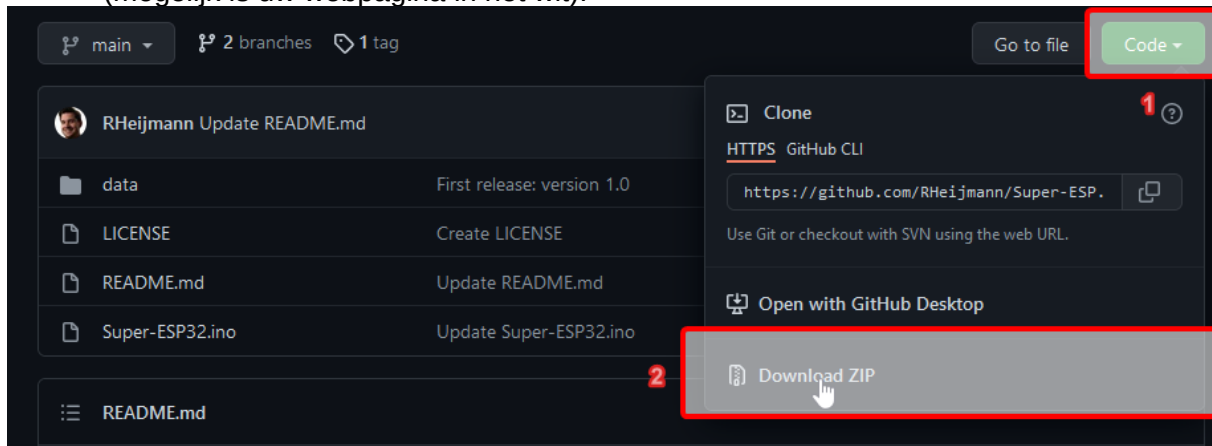


14. Klik op "OK".

15. Na een tijdje begint er een blauw lampje te schijnen naast het rode stroomlampje. Indien dit niet gebeurt neemt u contact op met uw contactpersoon.

4.1.2. Indien u enkel de configuratie wilt aanpassen

1. Navigeer naar: <https://github.com/RHeijmann/Super-ESP/>
Download hierbij de gehele GitHub-folder. Dit doet u door het volgende uit te voeren (mogelijk is uw webpagina in het wit):



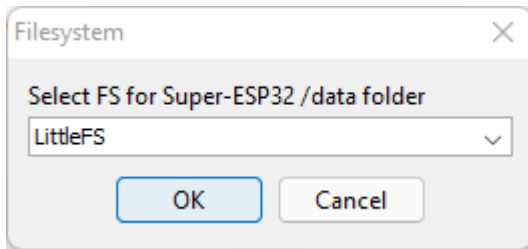
Figuur 3: Hoe download u een GitHub folder?

2. Extraheer het gedownloade zip-bestand.
3. Indien nog niet geïnstalleerd installeer: <https://www.arduino.cc/en/software>
4. Open vanuit de geëxtraheerde zip-bestand middels Arduino het bestand: Examples/Solarpanel/Super-ESP32.ino
5. Ben ervan verzekerd dat de volgenden boards zijn geïnstalleerd. Dit kunt u vanuit Arduino controleren door in de menubalk op "Tools" > "Board:...". Indien "Board:..." gelijk staat aan "Board: DOIT ESP32 DEVKIT V1" heeft u de juiste firmware geïnstalleerd. Zo niet dat klikt u op "File" > "Preferences". Hierin vult u bij het stukje "Additional Boards Manager URLs" het volgende in: https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_dev_index.json. Daarna klikt u op: "Tools" > "Board:..." > "Boards Manager". In het zoekvenster vult u "esp32" in. Installeer hier de 2.0.2 versie van "Espressif systems". Als laatste klikt u op "Tools" > "Board:..." > "ESP32 Arduino" > "DOIT ESP32 DEVKIT V1". Indien "Board:..." Nu heeft u de juiste firmware geïnstalleerd.
6. Open het "Examples/Solarpanel/data/configuration.json"-bestand in kladblok of een soort gelijke tekstbewerker.

7. Zoek en vervang de volgende waarden om uw apparaat te configureren:

| Zoek | Vervang met |
|--------------------------------|---|
| "hostname": "Solarpanel0001", | "hostname": "Solarpanel[Uw device ID]", |
| "ssid": "WiFinaam", | "ssid": "[Het SSID van uw WiFi-netwerk]", |
| "password": "WiFi-wachtwoord", | "password": "[Het wachtwoord van uw WiFi-netwerk]", |

8. Klik op "Tools" > "ESP32 Sketch Data Upload".
9. Selecteer "SPIFFS".



10. Klik op "OK".

11. Na een tijdje begint er een blauw lampje te schijnen naast het rode stroomlampje. Indien dit niet gebeurt neemt u contact op met uw contactpersoon.

4.2. Het plaatsen van het meetkastje

Benodigheden:

- Een geconfigureerd meetkastje, met daarbij alle benodigde sensoren voor uw dak,
- Een Micro USB-kabel met 5 volt stroom,
- Een 2.4 GHz 802.11n WiFi-verbinding met toegang tot het internet,
- 2 schroeven die geschikt zijn om in uw dak te boren,

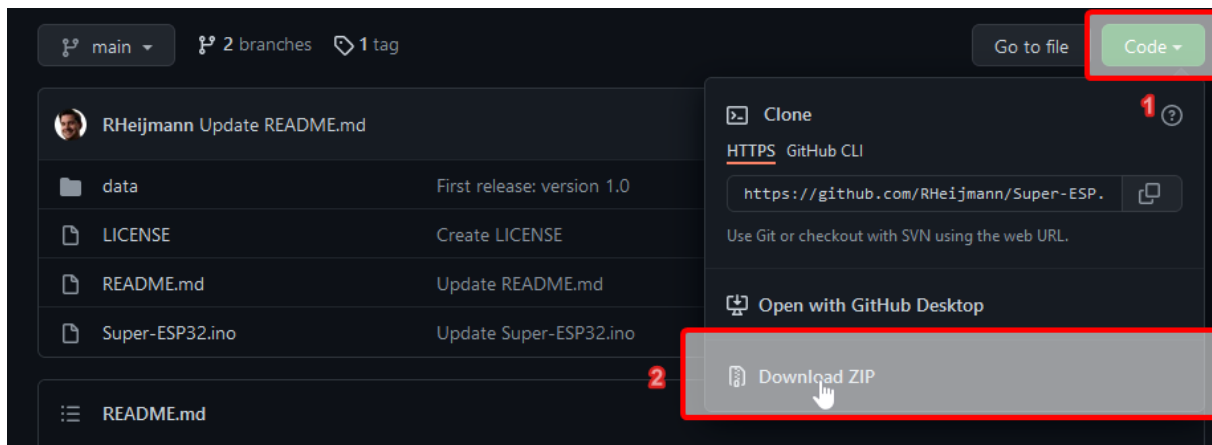
Plaats het meetkastje op het dak. Dit kan doormiddel van de 2 schroefgaten in de behuizing. Let hierbij op dat er een 5 volt spanningsbron aanwezig dient te zijn. Deze kan simpelweg aangevoerd worden via een extra zonnepaneel met accu of een enkele USB-kabel die over uw dak loopt.

Voor iedere zonnepaneel dient er een temperatuursensor en een stroomsensor aanwezig te zijn. Daarnaast dient er een extra temperatuursensor aanwezig te zijn voor het kunnen nemen van een nulmeting.

4.3. Het opstarten van de servers

Om uw meetgegevens in eigen handen te houden dient u het volgende te doen:

1. Installeer middels <https://www.docker.com/products/docker-desktop> Docker versie 4.3.2. Het installatie proces kan verschillen per besturingssysteem. De aangeleverde link Lost de meeste problemen op.
2. Indien u niet gebruik maakt van de Windows distributie van docker dient u ook "docker-compose" te installeren. Bekijk hiervoor: <https://docs.docker.com/compose/install/>
3. Indien u nog niet de firmware uit de eerdere instructies hebt gedownload
4. Download het docker-compose bestand door te navigeren naar: <https://github.com/RHeijmann/Super-ESP/>
Download hierbij de gehele GitHub-folder. Dit doet u door het volgende uit te voeren (mogelijk is uw webpagina in het wit):



Figuur 4: Hoe download u een GitHub folder?

2. Open “Examples/Solarpanel/Server/ElasticSearch Server and MQTT Broker/docker-compose.json” in kladblok of een soort gelijke tekstbewerker.
3. Zoek en vervang het volgende:

| Zoek | Vervang met |
|---|--|
| - GF_SECURITY_ADMIN_PASSWORD=0p3ns34rch | - GF_SECURITY_ADMIN_PASSWORD=[Uw wachtwoord voor het realtime visualisatie interface genaamd Grafana] |

4. Voer “docker-compose.json” uit door:
 - a. Linux: Naar “Examples/Solarpanel/Server/ElasticSearch Server and MQTT Broker/” te navigeren via een nieuw terminal venster. En vervolgens het commando “docker-compose up” uit te voeren.
 - b. Windows: Naar “Examples/Solarpanel/Server/ElasticSearch Server and MQTT Broker/” te navigeren met de Windows verkenner. En “Start.bat” uit te voeren.

Nu dat de servers online zijn is het tijd om de handler te draaien. Deze leest de MQTT-broker uit, controleert alle nieuwe meetwaarden en zet de waarden in de OpenSearch/ElasticSearch-database.

5. Installeer Python 3.10 via: <https://www.python.org/downloads/> . Of middels de gebruikelijke manier voor uw besturingssysteem.
6. Installeer de benodigde Python Libraries:
 - a. Linux: Naar “Examples/Solarpanel/Server/Handler” te navigeren via een nieuw terminal venster. En voer vervolgens het commando: pip install -r requirements.txt.
 - b. Windows: Naar “Examples/Solarpanel/Server/Handler/” te navigeren met de Windows verkenner. En “Install Dependencies.bat” uit te voeren.
7. Start vervolgens de handler door:
 - a. Linux: Naar “Examples/Solarpanel/Server/Handler” te navigeren via een nieuw terminal venster. En voer vervolgens het commando: python handler.py
 - b. Windows: Naar “Examples/Solarpanel/Server/Handler/” te navigeren met de Windows verkenner. En “Start.bat” uit te voeren.
8. Zet de volgende firewall poorten op uw router open:

| # | Afkomst | Bestemming | Afhandeling |
|---|---------|------------|-------------|
|---|---------|------------|-------------|

| | | | | |
|---|--------|-----------|---|--------------------------------------|
| 1 | *:9200 | DOCKER PC | Alleen toestaan indien vertrouwde bron. | OpenSearch Web API |
| 2 | *:9600 | DOCKER PC | Alleen toestaan indien vertrouwde bron. | OpenSearch Performance Monitor |
| 3 | *:5601 | DOCKER PC | Toestaan | OpenSearch Dashboards web interface. |
| 4 | *:3000 | DOCKER PC | Toestaan | Grafana web interface. |

9. Volg handleiding 4.1.2. en pas bij stap 7 ook het volgende aan:

| Zoek | Vervang met |
|--|--|
| "server": "predictivemaintenance.rik.blue", | "server": "[Het IP-adres van uw modem]", |

Gefeliciteerd als het goed is hebt u nu uw eigen servers draaidende. En ontvangt u data van alle meetkastje die u in stap 9 hebt geherconfigureerd.

Het standaard inlognaam en wachtwoord voor poort 9200, 9600 en 5601 is "admin". Voor Grafana (poort 3000) is de standaard inlognaam "admin" en het wachtwoord hiervan is door u tijdens stap 3 ingesteld.

4.1. Van database tot data scientist

De gebruikte databaseserversoftware heet OpenSearch (gebaseerd op Elasticsearch). Deze database is erg goed geïntegreerd met een van de meeste belangrijke data science tools: Python. De speciale python library genaamd "eland"¹ laat haar gebruikers de data uit de OpenSearch-database virtueel openen. Virtueel als in enkel de data die op dat moment nodig is wordt uit de database gehaald. Dit vermindert de druk op de database en geeft een snellere response uit. Zodra de datascientist de data heeft opgehaald, kan deze direct worden doorgezeten naar diverse machine learning en statistische analyse Python libraries.

Indien u de inloggegevens van de database hebt kunt u deze tool gebruiken om zelf aan de slag te gaan met uw data.

¹ <https://www.elastic.co/guide/en/elasticsearch/client/eland/current/overview.html>

```

>>> import eland as ed
>>> from elasticsearch import Elasticsearch

# First instantiate an 'Elasticsearch' instance connected to Elastic Cloud
>>> es = Elasticsearch(cloud_id="...", api_key=("...", "..."))

# then wrap the client in an Eland DataFrame:
>>> df = ed.DataFrame(es, es_index_pattern="flights")
>>> df.head(5)

```

| | AvgTicketPrice | Cancelled | ... | dayOfWeek | timestamp |
|---|----------------|-----------|-----|-----------|---------------------|
| 0 | 841.265642 | False | ... | 0 | 2018-01-01 00:00:00 |
| 1 | 882.982662 | False | ... | 0 | 2018-01-01 18:27:00 |
| 2 | 190.636904 | False | ... | 0 | 2018-01-01 17:11:14 |
| 3 | 181.694216 | True | ... | 0 | 2018-01-01 10:33:28 |
| 4 | 730.041778 | False | ... | 0 | 2018-01-01 05:13:00 |

```

[5 rows x 27 columns]

```

Figuur 5: Het ophalen van voorbeelddata uit Elasticsearch middels Eland.

4.1.1. Firewall regels