



Voor onze opdrachtgever de stad Herentals hebben wij van team Upsmart een smart systeem gebouwd voor openbare vuilnisbakken door aan deze vuilbakken sensoren en een microcontroller te monteren. Anderzijds toonden we in een proof-of-concept wat de mogelijkheden zijn van een openbare kiosk en maakten we een applicatie waarop de inwoner digitaal informatie kan bekijken en enquêtes kan invullen voor de stad Herentals.

Manual

[Project 4.0]

Jasper Guldentops, Seppe Alaerts, Sien Van Broekhoven, Matthias Thijsen, Yentl Stooabants, Koen Vorsters

Voorwoord

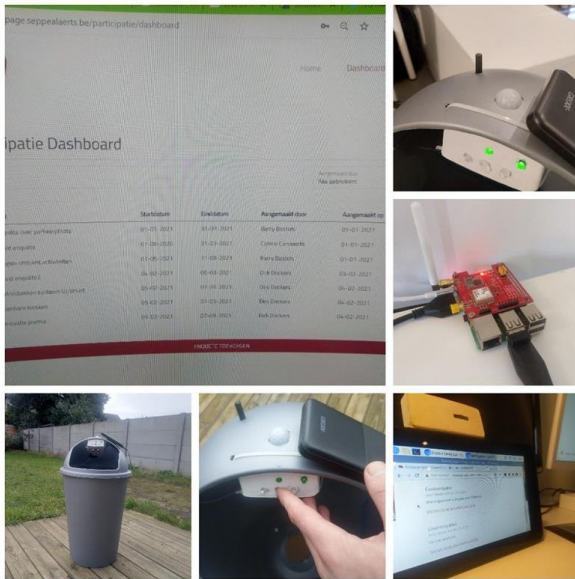
In het kader van onze opleiding Toegepaste Informatica en Elektronica-ICT aan de Thomas More hogeschool, kregen we tijdens het vak Project4.0 de kans om als team een smart city project voor Stad Herentals te realiseren. We willen daarom ook onze opdrachtgevers en docenten bedanken voor deze unieke kans en voor de ondersteuning tijdens de volledige cyclus van dit project.

Inhoud

Voorwoord	2
Voorstelling Project	5
Project vereisten	6
Smart vuilnisbaksysteem.....	6
Kiosk	6
Overzicht van het systeem	7
Hardware	8
Smart vuilnisbak	9
Schema:	9
Kiosk	13
Communicatie	14
LoRaWan:	14
LoRa gateway	15
The Things Network.	16
Code voor lora gateway	19
Cloud omgeving Azure	22
Overzicht	22
SQL DB	23
IOT central app	24
Azure IoT Central Device Bridge.....	25
TheThingsNetwork connectie	25
Azure Event Hub.....	26
Stream Analytics Task.....	28
Simulated Device Azure Central App	30
Data Kiosk.....	33
Twilio voor Azure (SMS berichten).....	34
Azure Defender (security)	40
Azure Defender for SQL.....	40
Azure Defender for Key Vault	40
Implementatie	40
Hosten Laravel backend op Azure.....	45
Web-app + MySQL.....	45
BI dashboards.....	50
Script.....	50
Datamodel	50

Visualisaties	51
Embedding	53
Applicatie.....	56
Overzicht	56
Opbouw	57
Installatie	57
Besluit.....	61

Voorstelling Project



In dit project hebben we een slim vuilnisbak systeem ontwikkeld voor onze opdrachtgever: stad Herentals. In ons prototype hebben we een device ontwikkeld met sensoren, batterij met zonnecel en een microcontroller en deze bevestigd aan een vuilnisbak. Zo kunnen we van de vuilnisbak een slim object maken. De data van de bevestigde sensoren wordt door de microcontroller verstuurd via het LoRaWan protocol naar The Things Network. Deze methode van communicatie maakt het mogelijk om op een energiezuinige manier sensordata te versturen over heel de stad. Via een aantal gateways die verbonden zijn met het internet kunnen we alle vuilnisbakken laten communiceren in de stad. Op het platform van

The Things Network converteren en stockeren we de data en sturen we deze data door naar de Microsoft Azure Cloud. Hier wordt dan alle data opgeslagen in een database.

Onze webapplicatie en Qlik sense dashboards linken we aan deze datasets. Hierdoor kunnen we deze data van het vuilnisbakkensysteem tonen en gebruiken in onze applicatie. Deze applicatie is dan ook een online tool voor de stad om zijn werknemers efficiënt het vuilnis te laten ophalen. Anderzijds is er ook een meldingssysteem geïntegreerd om sluikestorten te melden. Met een drukknop kunnen mensen aangeven of de plaats van de vuilbak netjes is of niet. Ook heeft de vuilnisbak een vuursensor aan boord. Indien er brand is gedetecteerd in de vuilnisbak zal het systeem een sms sturen naar de verantwoordelijke. Het belangrijkste aspect van deze slimme vuilnisbak is dat er bovenaan een afstandssensor constateert hoe veel de vuilnisbak is gevuld is. Ook is er onderaan een loadcell bevestigd die het inhoudelijk gewicht meet van de vuilnisbak.

In een tweede luik installeerde we een Raspberry Pi met een touchscreen voorzien van sensoren. Met deze opstelling wilden we onze opdrachtgever demonstreren welke mogelijkheden er zijn met een kiosk op openbare plaatsen. Voor deze kiosk maakte we een webapplicatie waar voorbijganger de mogelijkheid krijgen om enquêtes invullen voor de stad. Een werknemer van stad Herentals kan via de webapplicatie een enquête aanmaken en wijzigen. Ook maakten we van de kiosk een slim object dat we connecteren met de Azure Cloud om het aantal passanten en sensordata te versturen.

Project vereisten

Smart vuilnisbaksysteem

De proof-of-concept van het vuilnisbaksysteem moet aan de hand van sensoren meten hoe vol een vuilnisbak effectief is. Deze data zal dan naar een centrale database doorgestuurd worden.

Vervolgens zal een applicatie deze data uit de database uitlezen en presenteren op een gebruiksvriendelijke manier.

Wat zijn onze minimum vereisten hierbij:

- Ultrasonische sensor die de vullingsgraad van de vuilbak meet
- Een load cell meet het inhoudelijk gewicht
- Via een vuur (Infrarood) sensor detecteert men vuur/vlammen

Wat willen we extra uitwerken:

- De mogelijkheid om via een drukknop te kunnen melden (mail versturen) dat er naast de vuilnisbak gesluikstort is.
- Een bewegingssensor aan de buitenkant van de vuilnisbak om zo de trafiek aan passanten te kunnen meten.
- De mogelijkheid om in de app te kunnen bekijken wanneer een vuilnisbak door de stadsdiensten leeg gemaakt is.
- Wekelijks rapport via mail, verstuurd vanuit de app.
- Database-gegevens in Excel-formaat doorsturen vanuit de app (automatisch via mail)

Kiosk

Data, informatie en kennis zijn essentieel om (lokaal) beleid voor te bereiden en te ondersteunen. De kiosk is hierbij een meet- en leerinstrument voor zowel burgers als besturen. Elke inwoner kan zo helpen om het strategisch beleid van gemeenten en steden te analyseren, onderbouwen en sterker te maken!

Bij de uitwerking van dit proof-of-concept willen we eerst aangeven wat er niet zal gebeuren:

- Invullen, uitwerken of verantwoording van de eigenlijke fysieke kiosk op zich. De uitwerking van het concept moet volledig los kunnen staan van een fysieke plek.
- De uitwerking van een geregistreeerde aanmelding (via itsme of dergelijke)

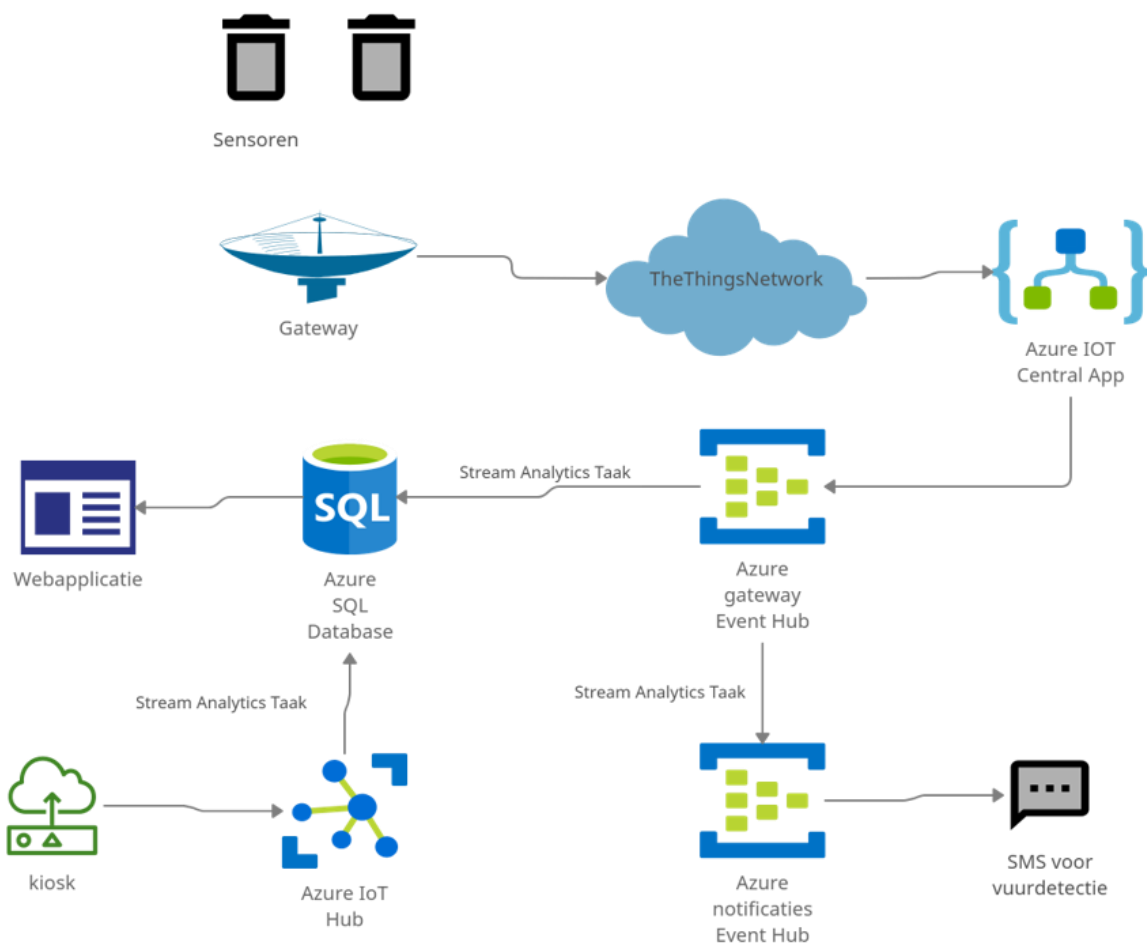
Het verwachte minimum bij de proof-of-concept:

- Gebruiksvriendelijke en 'dummy proof' interface
- Mogelijkheid om via een touchscreen vragen, peilingen en enquêtes te laten invullen door inwoners
- Flexibiliteit in de backend om de resultaten weer te geven (via grafieken, diagrammen,...)
- De vragen moeten aanpasbaar zijn

Wat willen we als extra uitwerken:

- Bewegingssensor die het aantal passanten registreert bij de kiosk

Overzicht van het systeem



In dit hoofdstuk willen wij u graag een overzicht geven van het gehele systeem dat we creëerden tijdens de realisatie fase. In bovenstaande afbeelding krijgt u een overzicht hoe het systeem juist functioneert. In een eerste fase realiseerden we de nodige hardware. Deze maakt het mogelijk de geregistreerde data van de kiosk en vuilnisbakken door te sturen naar de Microsoft Azure Cloud omgeving. Deze cloud-omgeving is dan ook het centrale punt waar de data word opgeslagen en de nodige data word opgehaald voor de applicatie en dashboards. Ook kunnen via deze omgeving functies aanmaken. Deze worden geactiveerd bij het overschrijden van waardes. Bijvoorbeeld bij het melden van brand in een vuilnisbak zal er een SMS verstuurd worden naar de verantwoordelijke. Ook maken wij gebruik van de beveiliging van deze cloud-omgeving om de data en onze applicatie te beschermen. In volgende hoofdstukken zullen wij dan ook dieper ingaan op deze deelsystemen en deze gedetailleerd beschrijven zodat u dit project eenvoudig kan opbouwen. In de projectmap kan u de benodigde files terugvinden. Indien u deze wenst te gebruiken zijn er een aantal aanpassingen in de codes noodzakelijk.

Hardware



In dit onderdeel zullen we dieper ingaan op de hardware. In een eerste stap willen we van onze vuilnisbak een slim object maken. Dit realiseerden we door enerzijds een microcontroller met sensoren te installeren in onze vuilnisbak en deze vervolgens te laten communiceren met ons systeem.

Zoals in het hoofdstuk “project vereisten” zijn de volgende componenten aangesloten aan de TTGO:

- Ultrasonische sensor die de vullingsgraad van de vuilbak meet
- Een load cell meet het inhoudelijk gewicht
- Via een vuur (Infrarood) sensor detecteert men vuur/vlammen
- Een drukknop om te melden dat er naast de vuilnisbak gesluikstort is.
- Een bewegingssensor aan de buitenkant van de vuilnisbak om zo de trafiek aan passanten te kunnen meten.
- Wanneer een vuilnisbak door de stadsdiensten leeg gemaakt is constateren door een magneet contact in het deksel te plaatsen.

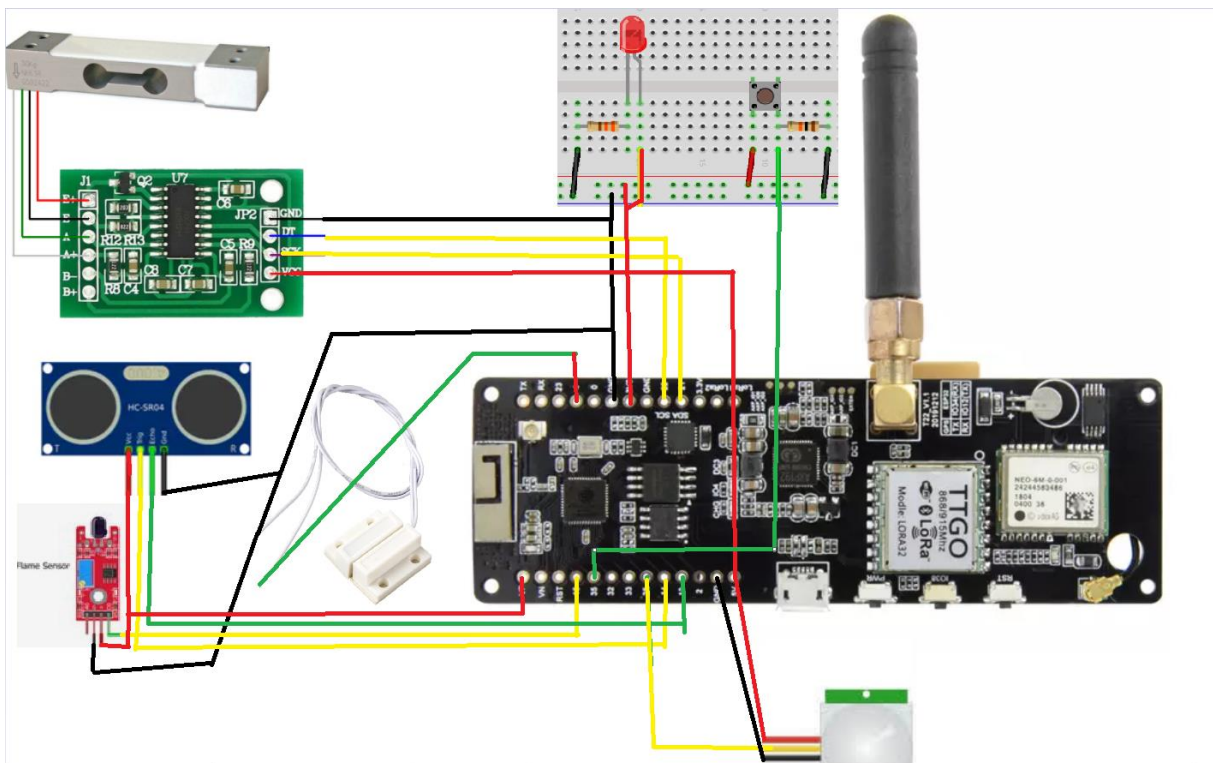
Smart vuilnisbak



In de eerste sprint bouwden we een proefopstelling op een breadboard. In de volgende sprint soldeerde we deze opstelling op een printplaat en 3D-printen we een case voor ons prototype. Deze monteerden we in een vuilnisbak. Zo kunnen we de inhoud en het gewicht van de vuilnisbak meten. Ook gebruiken we een vuursensor om te constateren of de vuilnisbak in brand staat. De drukknop vooraan maakt het mogelijk de melding te verzenden dat er werd gesluisstort. Al deze gegenereerde data zal dan verstuurd worden via een LoRa communicatie dat in het volgende hoofdstuk verder besproken wordt.

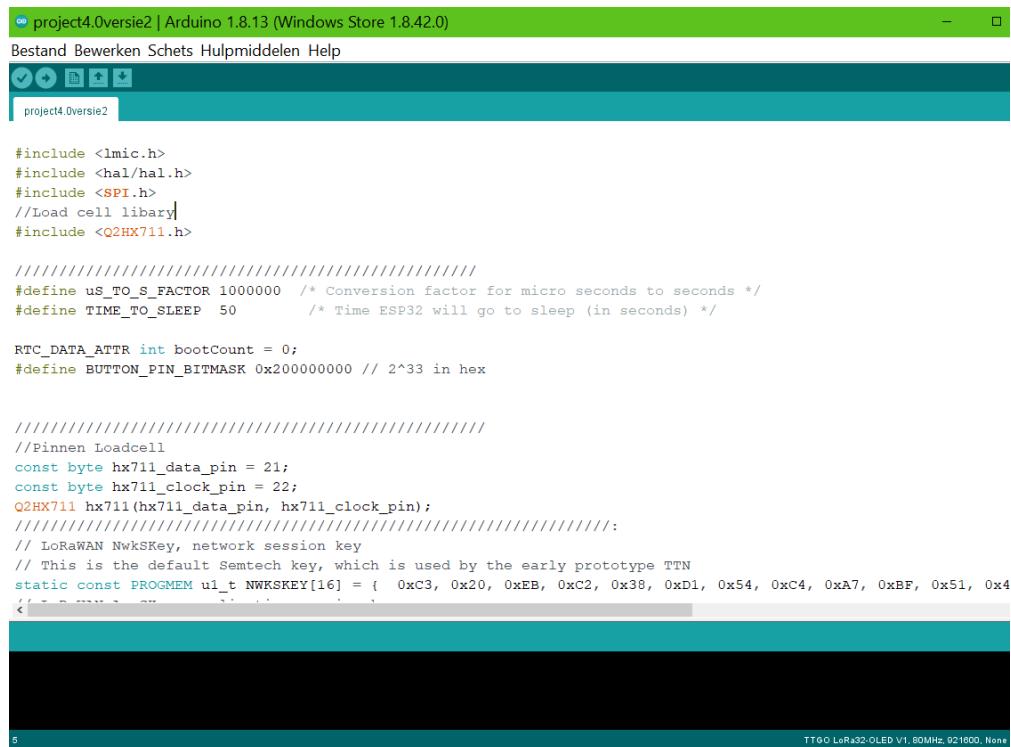
Als eerste worden de sensoren bevestigd volgens onderstaand schema aan de microcontroller.

Schema:



De software wordt via de Arduino IDE ingeladen in de microcontroller.

De "Println" in commentaar worden gebruikt om de controller te testen en de resultaten te bekijken in de terminal monitor van de Arduino software.



```
project4.0versie2 | Arduino 1.8.13 (Windows Store 1.8.42.0)
Bestand Bewerken Schets Hulpmiddelen Help

project4.0versie2

#include <lmic.h>
#include <hal/hal.h>
#include <SPI.h>
//Load cell library
#include <Q2HX711.h>

////////////////////////////////////
#define uS_To_S_FACTOR 1000000 /* Conversion factor for micro seconds to seconds */
#define TIME_TO_SLEEP 50 /* Time ESP32 will go to sleep (in seconds) */

RTC_DATA_ATTR int bootCount = 0;
#define BUTTON_PIN_BITMASK 0x200000000 // 2^33 in hex

////////////////////////////////////
//Pinnen Loadcell
const byte hx711_data_pin = 21;
const byte hx711_clock_pin = 22;
Q2HX711 hx711(hx711_data_pin, hx711_clock_pin);
////////////////////////////////////:
// LoRaWAN NwksKey, network session key
// This is the default Semtech key, which is used by the early prototype TTN
static const PROGMEM uint8_t NWKSEKEY[16] = { 0xC3, 0x20, 0xEB, 0xC2, 0x38, 0xD1, 0x54, 0xC4, 0xA7, 0xBF, 0x51, 0x4
```

De microcontroller zal dit programma doorlopen, de sensoren uitlezen en deze data versturen naar The Things Network.

In het eerste deel van de code worden de nodige libraries aangesproken. De lmich en hal/hal.h zijn ontwikkeld door IBM. Deze libraries maken het mogelijk de LoRa module te gebruiken en hiermee de berichten te versturen. De Q2HX711.h library zorgt voor de functionaliteit van de load cell. Zo kalibreren we deze bij opstart en lezen we het gewicht van de vuilnisbak in. Deze variabeel slaan we samen met de waardes van de andere sensoren op in een message.

```
#include <lmic.h>
#include <hal/hal.h>
#include <SPI.h>
//Load cell library
#include <Q2HX711.h>
```

```
////////////////////////////////////
```

```

First we configure the wake up source
We set our ESP32 to wake up every 5 seconds
*/
    esp_sleep_enable_ext0_wakeup(GPIO_NUM_33,1); //1 = High, 0 = Low
    esp_sleep_enable_timer_wakeup(TIME_TO_SLEEP * variabel);
    Serial.println("Setup ESP32 to sleep for every " + String(TIME_TO_SLEEP) +
    " Seconds");

    //esp_deep_sleep_pd_config(ESP_PD_DOMAIN_RTC_PERIPH, ESP_PD_OPTION_OFF);
    //Serial.println("Configured all RTC Peripherals to be powered down in sleep");

```

Daar we onze vuilnisbak willen voorzien van een batterij, is het noodzakelijk om zo veel mogelijk energie te besparen. Hiervoor maken we gebruik van een sleepfunctie op onze microcontroller. De timer en de knop op het device zal deze slaapstand onderbreken. Ook de pirsensor zal deze sleepfunctie onderbreken. Deze zal enkel de teller van het aantal passanten verhogen.

```

    }
}
// Checks if motion was detected, sets LED HIGH and starts a timer
void IRAM_ATTR detectsMovement() {
    Serial.println("MOTION DEECTED!!!");
    digitalWrite(led, HIGH);
    beweging =1;
    teller= teller + 1;
    startTimer = true;
    lastTrigger = millis();
}

```

Vervolgens zal de microcontroller om de 5 minuten de meetwaarde van sensoren en het aantal passanten doorsturen via het LoRa protocol. In onderstaande code kan u zien dat er een array wordt aangemaakt met alle variabelen van de sensoren, GPS coördinaten, teller (aantal passanten).

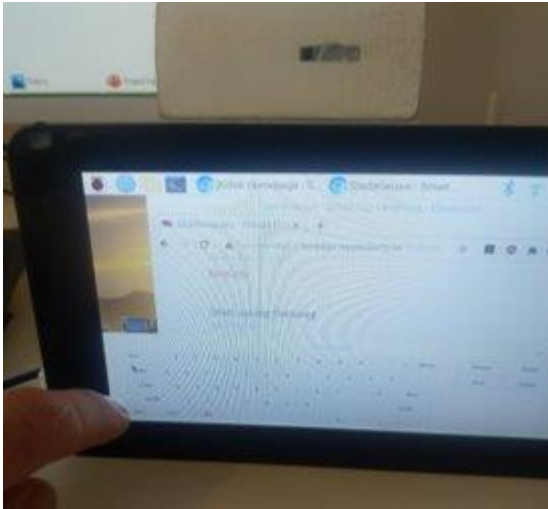
```

static uint8_t mydata[] = {gewicht, afstand, beweging, vuur, leeggemaakt, latitude, longitude, 76, sluikstort, teller,
// Serial.println(mydata);
beweging =0;
// Check if there is not a current TX/RX job running
if (LMIC.opmode & OP_TXRXPEND) {
    Serial.println(F("OP_TXRXPEND, not sending"));
} else {
    // Prepare upstream data transmission at the next possible time.
    LMIC_setTxData2(1, mydata, sizeof(mydata)-1, 0);
    Serial.println(F("Packet in queue gateway Lora"));
}
// Next TX is scheduled after TX_COMPLETE event.
}

void setup() {
    Serial.begin(115200);
}

```

Kiosk



Voor dit deelsysteem gebruiken we een Raspberry Pi 3B+. Deze sluiten we aan op een touchscreen. Via de GPIO pinnen sluiten we een PIR sensor en temp/vocht sensor aan. Deze data sturen we door naar de Azure Cloud omgeving. Om de applicatie te kunnen bedienen op de kiosk maken we gebruik van een Raspbian OS waar we de nodige gebruikers aanmaakten. We pasten de config files aan om de draairichting van het scherm te bepalen en het scherm te configureren. Bij opstart zal een Pythonscript in de achtergrond draaien. Ook werd er software geïnstalleerd om een keyboard op het touchscreen te gebruiken en het scherm te kalibreren. De gebruiker kan dit toestel gebruiken om voor de kiosk app.

In de achtergrond draait er een Python script op de kiosk dat het aantal bewegingen registreert (PIR-sensor). Elke 5 minuten zal het script het aantal passanten doorsturen samen met de gemeten temperatuur/vochtigheid. Al deze data wordt verstuurd naar het Azure platform om ook deze data te kunnen gebruiken in onze applicatie.

```
GNU nano 3.2 kiosk.py
import random
import Adafruit_DHT
import time
from azure.iot.device import IoTHubDeviceClient, Message
sensor = Adafruit_DHT.DHT11
pin = 4
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
PIR_PIN = 17
GPIO.setup(PIR_PIN, GPIO.IN)
aantal_mensen = int(0)
CONNECTION_STRING = "HostName=SmartCityIoT.azure-devices.net;DeviceId=kiosk;SharedAccessKey=0k0Cp2g+8SNjggjjhIhw47isziriuR8ovkIqlgdPY4="
MSG_SND = '{"temperature": (temperature),"humidity": (humidity),"aantal_mensen": (aantal_mensen)}'
```

```
pi@raspberrypi:~ $ sudo python3 kiosk.py
Motion Detected
Motion Detected
Motion Detected
Sending data to IoT Hub, press Ctrl-C to exit
Sending message: {"temperature": 21.0,"humidity": 49.0,"aantal_mensen": 3}
Message successfully sent
Motion Detected
```

Communicatie



LoRaWan:

LoRa staat voor 'long range' en is een specificatie voor een wide-area netwerk. Het is bedoeld voor devices die niet constant een internetverbinding nodig hebben, maar soms wat data moet doorsturen of ontvangen. Het netwerk heeft een bereik van tussen de 2,5 en 15 km per zendmast. De internetsnelheid ligt tussen de 0,3 en 50 kbit/s.

Deze technologie biedt veilige communicatie in de twee richtingen. Dankzij het lage stroomverbruik, lage kosten ten opzichte van andere technologieën en de mogelijkheid om veilige netwerken aan te leggen, is het LoRa-netwerk ideaal voor het internet of things.

De opbouw van LoRa bestaat uit nodes, gateways, netwerkserver en applicatieservers. Er worden radio frequency-chips gebruikt die een spread spectrum uitzenden. Een goed voorbeeld van een LoRa-netwerk is The Things Network.

Dit is een open netwerk en werd opgestart in Nederland. In minder dan een maand tijd voorzagen zij een dekking van heel de hoofdstad Amsterdam. De organisatie gelooft dat innovatie zal opbloeien als het netwerk volledig open en gratis is.

LoRa gateway



In vorig hoofdstuk hebben we uitgelegd dat de microcontrollers (nodes) sensordata versturen over LoRaWan. Deze communicatie zal worden opgevangen door (LoRa) gateways. Op een Raspberry Pi hebben we een LoRa expansion board geplaatst. Via het script worden de radiosignalen van onze hardware opgevangen en doorgestuurd via de internet verbinding op de Raspberry Pi naar het account van The Things Network.

In deze instructie video legt men uit hoe deze gateway geïnstalleerd wordt.

<https://www.youtube.com/watch?v=Ya-QIEaonLU&t=6s>

De afstand voor beide gateways in mijn woonplaats zijn groter dan 2,5 km, (interne antennes) daarom hebben we dan ook zelf onze gateway gebouwd zoals u in dit hoofdstuk kan lezen. In eerste plaats hebben wij een module besteld die op een Raspberry Pi kan worden gemonteerd. Nadien hebben wij via The Things Network deze gateway aangemaakt zodoende wij de nodige gegevens verkregen om in ons programma/script te plaatsen. In deze C-file hebben we dan een aantal aanpassingen moeten uitvoeren. Zoals het aantal berichten, zendtijd, locatie en de nodige keys. Daarna hebben we deze opgestart en kregen we via de website van The Things Network te zien dat deze gateway online was (zie onderstaande afbeelding). Zodra wij het gemaakte script van de node opstarten kregen wij via de webapplicatie te zien dat er data werd verstuurd. Door deze terug te converteren via de optie data payload konden wij de gegevens van onze vuilnisbak bekijken in de applicatie.

The Things Network.



In de eerste plaats moet er een account worden aangemaakt op The Things Network en word er een connectie gemaakt worden met de gateway. De gegenereerde keys moet u in de C file aanpassen op de gateway. Na deze file op te slaan compileer je dit bestand met het “make” commando in de terminal.

Gateways > eui-b6276b9f3e54a7

Overview Traffic Settings

GATEWAY OVERVIEW

Gateway ID: eui-b6276b9f3e54a7

Description: projectwireless_thomas_mone

Owner: koenvorstere [Transfer ownership](#)

Status: connected

Frequency Plan: Europe_868MHz

Router: bn-router-eu

Gateway Key: [key] [Reset](#)

Last Seen: 5 seconds ago

Received Messages: 349

Transmitted Messages: 0

INFORMATION

Brand: [dropdown]

Model: [dropdown]

Antenna: [dropdown]

LOCATION

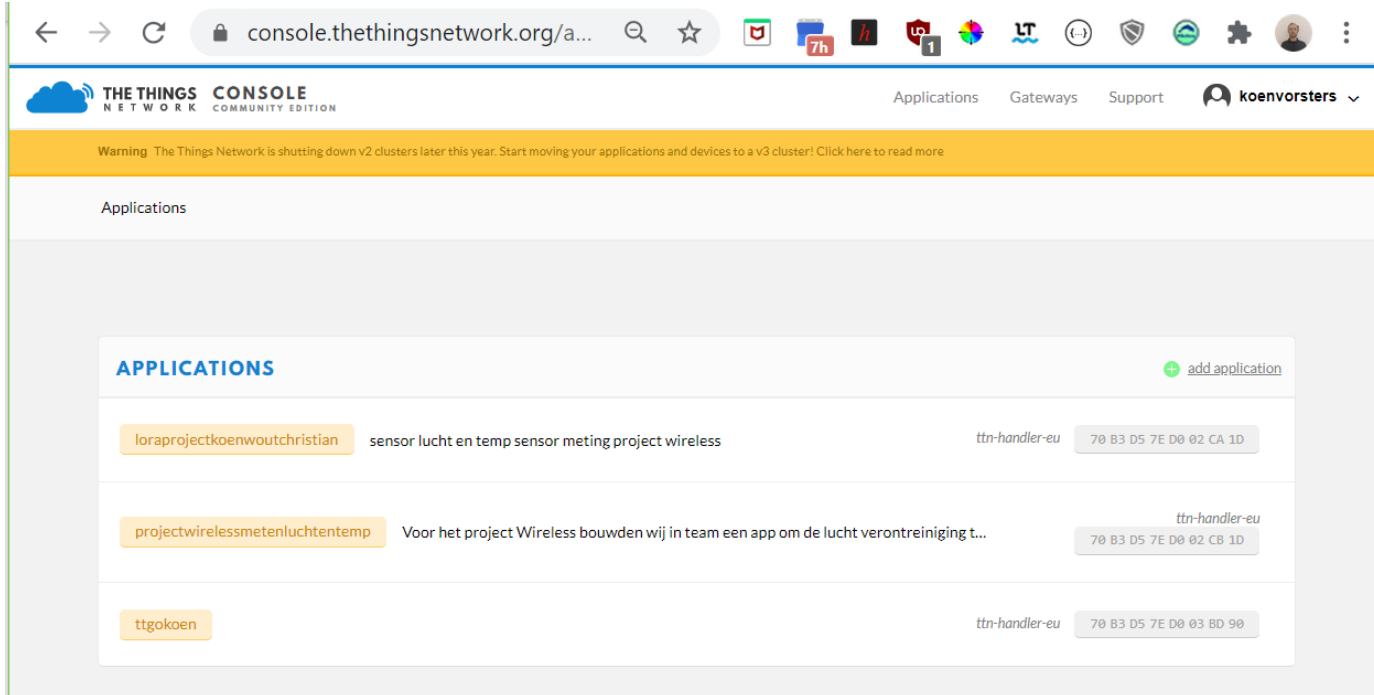
Antenna Placement: Indoor

Altitude: [dropdown]

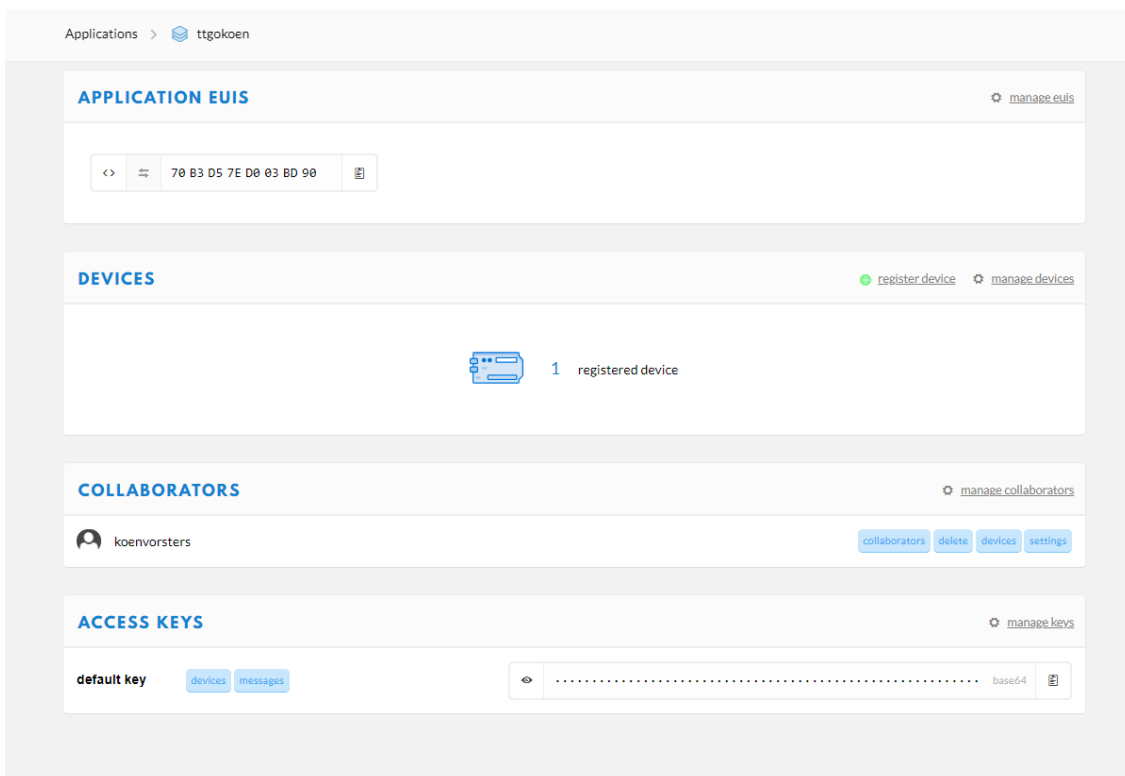
Map: [map] Lat: 52.367685, Lon: 4.8347622

Dit kan u controleren door het groene knopje bij de gateway in de applicatie. U kan ook connectie maken met andere gateways die beschikbaar zijn in uw locatie.

Vervolgens maakt u een applicatie aan.



Dan kan u de devices (node) aanmaken in de app door op add devices te klikken. Deze aangemaakte keys past u aan in de Arduino code. Zo heeft elk device een unieke code en ID en zorgen deze keys voor een beveiliging van de data al vanaf het doorsturen over het LoRa network.



Om te controleren dat het systeem tot deze stap functioneert testen we het systeem uit. We laten de microcontroller berichten versturen naar The Things Network. Als dit succesvol is zal u de verzonden

data ontvangen in de payload van The Things Network.

Applications > ttgokoer > Payload Formats

Overview Devices **Payload Formats** Integrations Data Settings

PAYLOAD FORMATS

Payload Format
The payload format sent by your devices

Custom

decoder converter validator encoder [remove decoder](#)

```
1 function Decoder(bytes, port) {  
2  
3   var gewicht = bytes[0];  
4   var afstand = bytes[1];  
5   var beweging = bytes[2];  
6   var vuur = bytes[3];  
7   var leeggemaakt = bytes[4];  
8   var lat = bytes[5];  
9   var lon = bytes[7];  
10  var sluikstorten = bytes[8];  
11  var teller = bytes[9];  
12  var send_message = bytes[10];  
13}
```

decoder has no changes

Payload

0 bytes 1 Test

Om deze data te kunnen versturen naar Azure maken we een integratie aan. Deze integratie zal al de data van The Things Network doorsturen naar de cloud. Hiervoor hebben we een url en de beveiligingskey van Azure nodig.

In dit filmpje kan u bekijken hoe deze integraties juist werkt en hoe deze word geïntegreerd.

https://www.youtube.com/watch?v=Uebcq7xml1M&index=2&list=PLM8eOeiKY7JVwrBYRHxsf9p0VM_dVapXI


Applications > ttgokoen > Integrations > dev1

Overview Devices Payload Formats **Integrations** Data Settings

INTEGRATION OVERVIEW

Process ID dev1

Status ● Running

Platform  HTTP Integration (v2.6.0) [documentation](#)

Author The Things Industries B.V.

Description Sends uplink data to an endpoint and receives downlink data over HTTP.

SETTINGS

Access Key
The access key used for downlink

default key devices messages

URL
The URL of the endpoint

Na deze investering in de nodige gateways en infrastructuur zijn deze frequentie banden gratis te gebruiken. Wel in regel zijnde met de policy van The Things network. Een zeer belangrijk gegeven hier is de “airtime” per device. Bij een uitrol van alle vuilnisbakken zal hier de berekening gemaakt op moeten worden “de tijd dat een device een bericht kan versturen”. Zo is het dan ook mogelijk om duizenden vuilnisbakken (nodes) te laten communiceren over dit netwerk. Hetzij beperkt in data, toch het enorme voordeel om op een energie zuinige manier een zeer groot bereik te hebben om data te verzenden en ontvangen.

Code voor lora gateway

```
#include <string>
#include <stdio.h>
#include <sys/types.h>
#include <sys/ioctl.h>
#include <net/if.h>
#include "base64.h"
#include <wiringPi.h>
#include <wiringPiSPI.h>
typedef bool boolean;
typedef unsigned char byte;
static const int CHANNEL = 0;
byte currentMode = 0x81;
char message[256];
char b64[256];
bool sx1272 = true;
byte receivedbytes;
struct sockaddr_in si_other;
int s, slen=sizeof(si_other);
struct ifreq ifr;
```

...

* Deze waardes moet je aanpassen afhankelijk van de pin out van de lora module !

```

*
*****/
// SX1272 - Raspberry connections
int ssPin = 6;
int dio0 = 7;
int RST = 0;
// Set spreading factor (SF7 - SF12)
sf_t sf = SF7;
// Set center frequency deze frequentie word gebruikt in europa
uint32_t freq = 868100000; // in Mhz! (868.1)
// Set location de locatie die doorgestuurd word naar the Thingsnetwork
float lat=51.32919197;
float lon=4.95441139;
int alt=30;
/* Informal status fields */
static char platform[24] = "Single Channel Gateway"; /* platform definition */
static char email[40] = "koenvorst@ gmail.com"; /* used for contact email */
static char description[64] = "koenvorst"; /* used for free form description */
// define servers
// TODO: use host names and dns
#define SERVER1 "52.169.76.203" // The Things Network: croft.thethings.girotito.nl
// #define SERVER2 "192.168.1.10" // local
#define PORT 1700 // The port on which to send data
// #####
// #####
...

buff_up[4] = (unsigned char)ifr.ifr_hwaddr.sa_data[0];
buff_up[5] = (unsigned char)ifr.ifr_hwaddr.sa_data[1];
buff_up[6] = (unsigned char)ifr.ifr_hwaddr.sa_data[2];
buff_up[7] = 0xFF;
buff_up[8] = 0xFF;
buff_up[9] = (unsigned char)ifr.ifr_hwaddr.sa_data[3];
buff_up[10] = (unsigned char)ifr.ifr_hwaddr.sa_data[4];
buff_up[11] = (unsigned char)ifr.ifr_hwaddr.sa_data[5];

/* start composing datagram with the header */
uint8_t token_h = (uint8_t)rand(); /* random token */
uint8_t token_l = (uint8_t)rand(); /* random token */
buff_up[1] = token_h;
buff_up[2] = token_l;
buff_index = 12; /* 12-byte header */

// TODO: tmst can jump is time is (re)set, not good.
struct timeval now;
gettimeofday(&now, NULL);
uint32_t tmst = (uint32_t)(now.tv_sec*1000000 + now.tv_usec);

/* start of JSON structure */

```

```

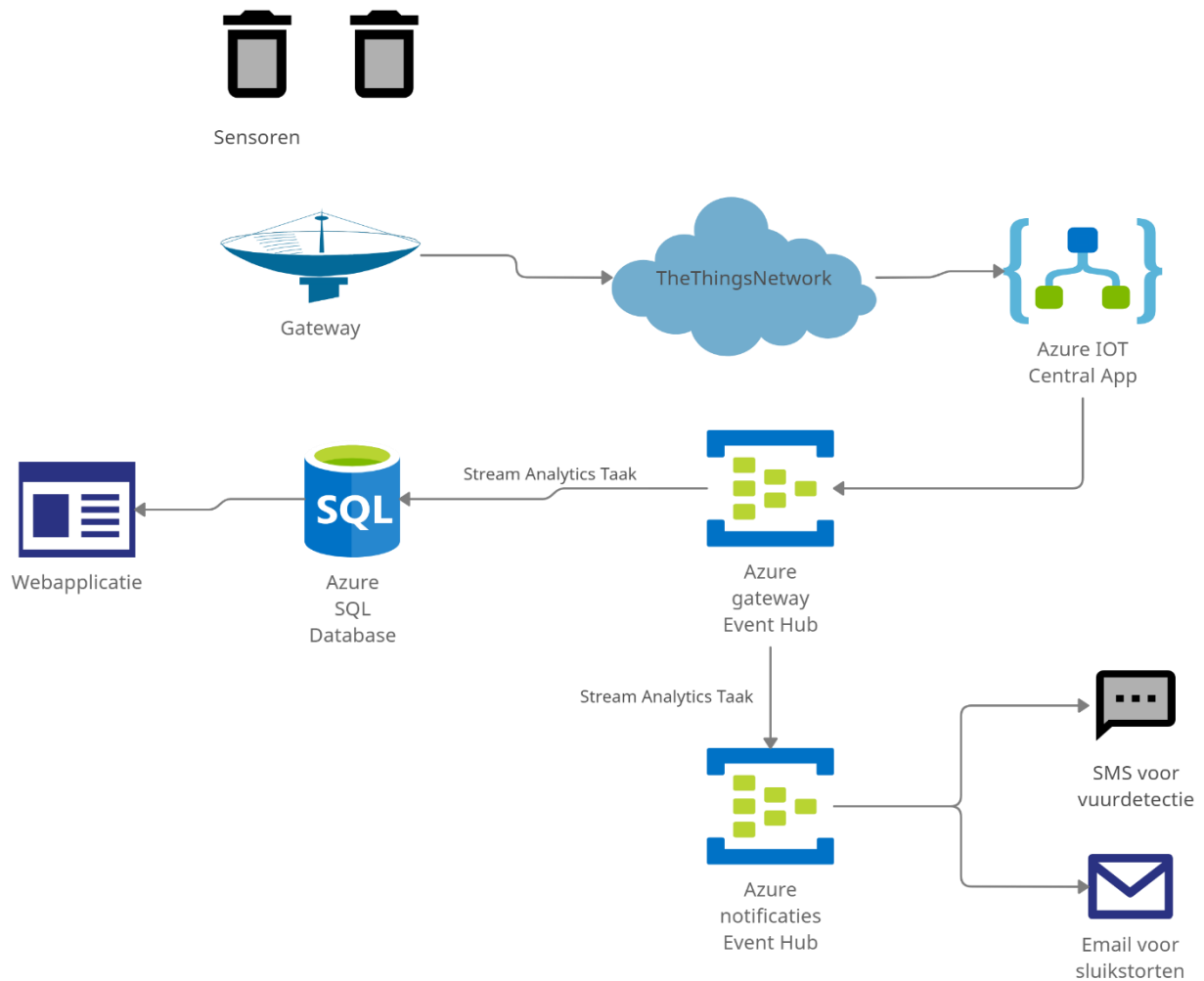
memcpy((void *)(buff_up + buff_index), (void *)"\rxpk\":[", 9);
buff_index += 9;
buff_up[buff_index] = '{';
++buff_index;
j = snprintf((char *)(buff_up + buff_index), TX_BUFF_SIZE-buff_index, "\"tmst\":%u", tmst);
buff_index += j;
j = snprintf((char *)(buff_up + buff_index), TX_BUFF_SIZE-buff_index,
",\"chan\":%1u,\"rfch\":%1u,\"freq\":%.6lf", 0, 0, (double)freq/1000000);
buff_index += j;
memcpy((void *)(buff_up + buff_index), (void *)",\"stat\":1", 9);
buff_index += 9;
memcpy((void *)(buff_up + buff_index), (void *)",\"modu\":\"LORA\"", 14);
buff_index += 14;
/* Lora datarate & bandwidth, 16-19 useful chars */
switch (sf) {
case SF7:
    memcpy((void *)(buff_up + buff_index), (void *)",\"datr\":\"SF7", 12);
    buff_index += 12;
    break;
case SF8:
    memcpy((void *)(buff_up + buff_index), (void *)",\"datr\":\"SF8", 12);
    buff_index += 12; ...

```

Cloud omgeving Azure

Overzicht

De Azure omgeving is geconfigureerd volgens onderstaand schema.



De Azure IoT Central Device Bridge maakt het mogelijk om data van externe bronnen, zoals TheThingsNetwork, in Azure services te integreren. Via de Azure IOT Central App kunnen we onze apparaten managen en ruwe log data bekijken.

Vervolgens wordt deze data doorgestuurd naar een Azure Event Hub waar deze data een dag in bijgehouden wordt. De bedoeling van deze event hub is om data in real time van onze Central App naar onze database te kunnen sturen.

De stream analytics taak die op de SQL database geconfigureerd is, gaat wanneer er nieuwe data beschikbaar is in de event hub, deze in real-time wegschrijven naar de SQL database.

Vervolgens wordt de data ook nog naar een notificaties event hub geschreven, deze is bedoeld om de data die triggers genereert voor bijvoorbeeld SMS berichten wanneer er vuur is, bij te houden.

SQL DB

Stap 1 is een SQL database aanmaken op het Azure portal.

<https://portal.azure.com/#create/Microsoft.SQLDatabase>

Microsoft Azure

Startpagina > Nieuw > SQL Database >

SQL-database maken

Microsoft

Basis Netwerken Aanvullende instellingen Tags Beoordelen en maken

Een SQL-database maken met uw voorkeursconfiguraties. Vul het tabblad Basisinformatie in en ga vervolgens naar Beoordelen en maken om in te richten met slimme standaardinstellingen of pas instellingen aan op elk tabblad. [Meer informatie](#)

Projectgegevens

Selecteer het abonnement om geïmplementeerde resources en kosten te beheren. Gebruik resourcegroepen als mappen om al uw resources te organiseren en beheren.

Abonnement

Resourcegroep

Databasedetails

Voer de vereiste instellingen voor deze database in, waarbij u een logische server kiest en de resources voor berekening en opslag configureert.

Databasenaam

Server

Wilt u een elastische pool voor SQL gebruiken? ☐ Ja ☒ Nee

Berekening en opslag

Netwerken

Toestaan dat Azure-services en -resources toegang tot deze server krijgen ☒

Huidig IP-adres van client toevoegen ☒

Privé-eindpunt ☐

Beoordelen en maken **Volgende: Netwerken >**

Het type database is: Standard S1 – 20 DTU, 250GB

Microsoft Azure

Startpagina > Nieuw > SQL Database >

SQL-database maken

Microsoft

Basis Netwerken Aanvullende instellingen Tags Beoordelen en maken

Productdetails

SQL-database door Microsoft

Geschatte kosten per maand 25.30 EUR

Gebruiksrechtsovereenkomst

Door op Maken te klikken, (a) ga ik akkoord met de juridische voorwaarden en privacyverklaring(en) die zijn gekoppeld aan de bovenstaande Marketplace-aanbieding(en); (b) geef ik Microsoft toestemming om kosten die zijn gekoppeld aan de aanbieding(en), te factureren volgens mijn huidige betalingswijze met dezelfde factureringsfrequentie als voor het Azure-abonnement; en (c) ga ik ermee akkoord dat Microsoft mijn contact-, gebruiks- en transactiegegevens kan delen met de provider(s) van de aanbieding(en) voor ondersteunings-, facturerings- en andere transactieactiviteiten. Microsoft verstrekt geen rechten op aanbiedingen van externe leveranciers. Raadpleeg voor verdere details de [voorwaarden voor Azure Marketplace](#).

Basis

Abonnement Visual Studio Enterprise-abonnement

Resourcegroep rg_SmartCity08

Regio West-Europa

Databasenaam SmartCity08

Server (nieuw) smartcity-herentals

Berekening en opslag Standard S1: 20 DTU's, 250 GB opslag

Netwerken

Toestaan dat Azure-services en -resources toegang tot deze server krijgen ☒

Huidig IP-adres van client toevoegen ☒

Privé-eindpunt ☐

Aanvullende instellingen

Maken **Volgende** Een sjabloon voor automation downloaden

Nieuwe tabellen kunnen aangemaakt worden met onderstaand script in de query editor.

```
CREATE TABLE Data
```

```
(
```

```
    Deviceld INT IDENTITY PRIMARY KEY,
```

```
    Name NVARCHAR(128) NOT NULL
```


);

IOT central app





<https://smartcity-herentals.azureiotcentral.com/dashboards/urn:homepageView:els1mecc>

<https://docs.microsoft.com/en-us/azure/iot-central/core/quick-deploy-iot-central>

Vervolgens maken we een IOT Central app aan op <https://apps.azureiotcentral.com/>.

 **Azure IoT Central**

Over uw app



Toepassingsnaam * ⓘ

SmartCity Herentals

URL * ⓘ

smartcity-herentals.azureiotcentral.com

Toepassingsjabloon * ⓘ

Aangepaste toepassing

Prijsplan *

☐ **Gratis**
Probeer 7 dagen lang zonder verplichtingen
5 gratis apparaten

☐ **Standard 0**
Voor apparaten die enkele berichten per dag verzenden
2 gratis apparaten 400 berichten/mnd

☐ **Standaard 1**
Voor apparaten die berichten per uur verzenden
2 gratis apparaten 5.000 berichten/maand

☒ **Standaard 2 (meest gebruikt)**
Voor apparaten die om de paar minuten berichten verzenden
2 gratis apparaten 30.000 berichten/maand

Factureringsgegevens

Map * ⓘ

Student Ambassadors (stdntpartners.onmicrosoft.com)

Azure-abonnement * ⓘ

Hebt u geen abonnement? [Abonnement maken](#)

Visual Studio Enterprise-abonnement

Locatie * ⓘ

Europa

Prijs Standard tier 2 = 0.59 euro per maand, 30K messages per device. (1 message = 4Kb).

Azure IoT Central Device Bridge

Aan de hand van een template:

<https://portal.azure.com/#create/Microsoft.Template/uri/https%3A%2F%2Fraw.githubusercontent.com%2FAzure%2FIotc-device-bridge%2Fmaster%2Fazuredeploy.json>



Startpagina >



Aangepaste implementatie

Implementeren vanuit een aangepast sjabloon

Grondbeginselen Beoordelen en maken

Sjabloon

 Aangepaste sjabloon  4 resources

 Sjabloon bew...  Parameters b...

Implementatiebereik

Selecteer het abonnement om geïmplementeerde resources en kosten te beheren. Gebruik resourcegroepen als mappen om als uw resources te beheren.

Abonnement * ⓘ

Resourcegroep * ⓘ [Nieuw](#)

Parameters

Regio * ⓘ

Scope ID * ✓

IoT Central SAS Key * ✓

Installeren npm in app services console en restarten.

```
C:\home\site\wwwroot\IoTIntegration>npm install
audited 54 packages in 1.609s
found 5 low severity vulnerabilities
run `npm audit fix` to fix them, or `npm audit` for details
```

TheTingsNetwork connectie

HTTP post requests moeten vanaf TTN naar deze link gestuurd worden <https://iotc-fn4wnefq2ezf2tm.azurewebsites.net/api/IoTCIntegration?code=il/eRHbm2CN0peBXazFW2CCa4kr6d9lp5eKeo6ahPnlcN5DK/na/Ow==>

Body format template:

```
{
  "device": {
```

```
    "deviceId": "my-cloud-device"
  },
  "measurements": {
    "temp": 20.31,
    "pressure": 50,
    "humidity": 8.5,
    "ledColor": "blue"
  }
}
```

Measurements objectnamen moeten overeenkomen met namen van device template in IoT central app indien een template gebruikt wordt.

Voeg code toe voor “handleMessage” op lijn 21 in IoTIntegrations/index.js in azure functie.

```
req.body = {
  device: {
    deviceId: req.body.hardware_serial.toLowerCase()
  },
  measurements: req.body.payload_fields
};
```

Azure Event Hub

<https://techcommunity.microsoft.com/t5/azure-iot/exporting-iot-central-data-to-azure-sql-database/m-p/1347787>

<https://www.codeproject.com/Articles/1169531/Sending-events-from-Azure-Event-Hub-to-Azure-SQL-D>

Namespace aanmaken.

Startpagina >

Naamruimte maken

Event Hubs

Basisinformatie Functies Tags Beoordelen en maken

PROJECTGEGEVENS

Selecteer het abonnement om geïmplementeerde resources en kosten te beheren. Gebruik resourcegroepen als mappen om als uw resources te beheren.

Abonnement * Visual Studio Enterprise-abonnement ▼

Resourcegroep * (Nieuw) rg-SmartCityEventHub ▼

[Nieuw](#)

INSTANTIEGEGEVENS

Voer de vereiste instellingen voor deze naamruimte in, inclusief een prijscategorie en de configuratie van het aantal doorvoereenheden.

Naam van naamruimte * SmartCityHerentals ✓ .servicebus.windows.net

Locatie * West-Europa ▼

Prijscategorie (Volledige prijsgegevens weergeven) * Basis (1 Consumentengroep, 100 Brokered verbindingen) ▼

Doorvoereenheden * 3

Event hub aanmaken binnen namespace.

Startpagina > SmartCityHerentals >

Event Hub maken

Event Hubs

Naam * ⓘ ToSQLGateway ✓

Aantal partities ⓘ 2

Bewaarperiode van bericht ⓘ 1

Vastleggen ⓘ Aan Uit

App toegang geven in Instellingen/Beleid voor gedeelde toegang, key kopiëren.

tosqlgateway (SmartCityHerentals/tosqlgateway) | Beleid voor gedeelde toegang

Event Hubs-instantie

Zoeken (Ctrl+/) « + Toevoegen

Overzicht

Toegangsbeheer (IAM)

Problemen vaststellen en oplossen

Instellingen

Beleid

AppToegang

Claims

Verzenden

In app een nieuwe export configureren naar Event Hub.

Nieuwe bestemming

Als u een nieuwe bestemming wilt maken, selecteert u een doeltype en voert u de verbindingsgegevens in.
[Meer informatie](#)

Doelnaam *

Doeltype *

Azure Event Hubs

Verbindingsreeks * ⓘ

Event Hub ⓘ

* Vereist

MakenAnnuleren

Stream Analytics Task

<https://docs.microsoft.com/en-us/azure/azure-sql/database/stream-data-stream-analytics-integration>

Nieuwe stream analytics taak maken in SQL Database.

Maken

Vernieuwen

Feedback

Stream Analytics-taken

Taken zoeken...

Naam <div></div>	Invoer <div></div>	Uitvoer <div></div>	Locatie <div></div>	Status <div></div>
Data	gatewaysql	DataSensoren	West Europe	<div></div> Actief

Bij invoer de Event hub selecteren, bij uitvoer de tabel in de database.

Vervolgens de query editen om specifieke data uit de event hub te lezen en naar de database te schrijven.

Query voor data van fysiek apparaat:

```
SELECT
    deviceId as ID,
    DATEADD(hh, 1, enqueuedTime) as tijd,
    51.178013 AS latitude,
    4.840502 AS longitude,
    telemetry.afstand AS afstand,
    telemetry.beweging AS beweging,
    telemetry.gewicht AS gewicht,
    telemetry.leeggemaakt AS leeggemaakt,
    telemetry.sluikestorten AS sluikestorten,
    telemetry.vuur AS vuur,
    telemetry.teller AS aantal_sluikestorten
```

```

INTO
[DataSensoren]
FROM
[gatewaysql]
WHERE
deviceId = '004e6fb48a69387d'

```

Query opslaan en Stream analytics taak starten.


Resultaat

Sensorgegevens worden in real-time doorgestuurd naar SQL Database:

ID	tijd	afstand	beweging	gewicht	leegemaakt	vuur	longitude	latitude	sluikstorten	aantal_sluike...
004e6fb48a6938...	29-1-2021 13:29:56	48	1	1	1	0	4.840502	51.178013	1	1
004e6fb48a6938...	29-1-2021 13:15:14	48	1	1	0	0	4.840502	51.178013	1	1
004e6fb48a6938...	29-1-2021 13:09:44	48	1	1	0	0	4.840502	51.178013	1	1
004e6fb48a6938...	29-1-2021 12:58:31	48	1	1	0	0	4.840502	51.178013	1	1
004e6fb48a6938...	29-1-2021 12:34:49	48	1	1	0	0	4.840502	51.178013	1	1
004e6fb48a6938...	29-1-2021 12:34:45	48	1	1	0	0	4.840502	51.178013	1	1
004e6fb48a6938...	29-1-2021 12:34:36	48	1	1	0	0	4.840502	51.178013	1	1
004e6fb48a6938...	29-1-2021 12:34:24	48	1	1	0	0	4.840502	51.178013	1	1
004e6fb48a6938...	29-1-2021 12:04:02	48	1	1	0	0	4.840502	51.178013	1	1
004e6fb48a6938...	29-1-2021 11:04:59	48	1	1	0	0	4.840502	51.178013	1	1
004e6fb48a6938...	29-1-2021 11:04:39	48	1	1	0	0	4.840502	51.178013	1	1
004e6fb48a6938...	29-1-2021 11:04:23	48	1	1	0	0	4.840502	51.178013	1	1
004e6fb48a6938...	29-1-2021 11:04:15	48	1	1	0	0	4.840502	51.178013	1	1
004e6fb48a6938...	29-1-2021 11:04:03	48	1	1	0	0	4.840502	51.178013	1	1
004e6fb48a6938...	29-1-2021 11:03:00	48	1	1	0	0	4.840502	51.178013	1	1
004e6fb48a6938...	29-1-2021 11:02:29	48	1	1	0	0	4.840502	51.178013	1	1

Simulated Device Azure Central App

Device template aanmaken



Apparaatsjablonen > SimDev v7 > **Model**

SimDev v7

De toepassing is bijgewerkt: 23 uur geleden De interfaces zijn gepubliceerd: 4 dagen geleden

Model

SimDev

Cloudeigenschappen

Aanpassen

Weergaven

Onbewerkte gegevens

+ Onderdeel toevoegen


→ Exporteren

🗑 Verwijderen

aantalsluikstorten	aantalsluikstorten	Telemetrie
afstand	afstand	Telemetrie
beweging	beweging	Telemetrie
gewicht	gewicht	Telemetrie
lat	lat	Telemetrie
leeggemaakt	leeggemaakt	Telemetrie
lon	lon	Telemetrie
sluikstorten	sluikstorten	Telemetrie
vuur	vuur	Telemetrie

Minimum en maximum waarden aanpassen in Model/Aanpassen.

Nieuw simulated device



Alle apparaten

+ Nieuw

← Importeren

→ Exporteren

📄 Goedkeuren

🕒 Blokkeren

⏏ Blokkering opheffen


🔗 Koppelen aan gateway

🔄 Migreren

⋮

Apparaatnaam	Apparaat-id	Gesimuleerd	Apparaatstatus	Apparaatsjabloon
jq3twyzk5g	jq3twyzk5g	Ja	Geblokkeerd	SimDev v7

Apparaten / SimDev v7 / jq3twyzk5g



jq3twyzk5g

Onbewerkte gegevens

Tijdstempel ↓	Berichttype	afstand	beweging
✓ 1-2-2021 13:34:43	Telemetrie	27	1

Select a node...

```

▼ object {11}
  _eventtype : Telemetrie
  _timestamp : 2021-02-01T12:34:43.671Z
  aantalsluikstorten : 2
  afstand : 27
  beweging : 1
  gewicht : 2719
  lat : 51.331281
  leeggemaakt : 0
  lon : 4.945604
  sluikstorten : 0
  vuur : 0
  
```

Data doorsturen naar Azure Event hub door export functie naar SQLgateway event hub.

Event hub naar SQL DB

<https://docs.microsoft.com/en-us/azure/azure-sql/database/stream-data-stream-analytics-integration>

<https://www.codeproject.com/Articles/1169531/Sending-events-from-Azure-Event-Hub-to-Azure-SQL-D>

In DB - Nieuwe stream analytics taak:

```

SELECT
  deviceId as ID,
  DATEADD(hh, 1, enqueuedTime) as tijd,
  51.171092 AS latitude,
  4.832242 AS longitude,
  telemetry.afstand AS afstand,
  telemetry.beweging AS beweging,
  telemetry.gewicht AS gewicht,
  telemetry.leeggemaakt AS leeggemaakt,
  telemetry.sluikstorten AS sluikstorten,
  telemetry.aantalsluikstorten AS aantal_sluikstorten,
  telemetry.vuur AS vuur

```

INTO

```

[DataSensoren]
FROM
[gatewaysql]
WHERE
deviceId = 'jq3twyzk5g'

```

Start stream analytics taak

Stream Analytics-taak starten

SIM

i Start deze Stream Analytics-taak om streaminggegevens vanuit Event Hub/IoT Hub in een SQL-tabel op te nemen.

Starttijd uitvoer ⓘ
☒ Nu ☐ Aangepast

Streaming-eenheden ⓘ
 6

Foutafhandeling voor uitvoergegevens ⓘ
 Opnieuw proberen

Uitvoerinstellingen voor SQL Database
☒ Alle invoerpartities samenvoegen tot één schrijver
☐ Partitieschema van vorige querystap overnemen of invoeren

Maximum aantal batches ⓘ
 10000

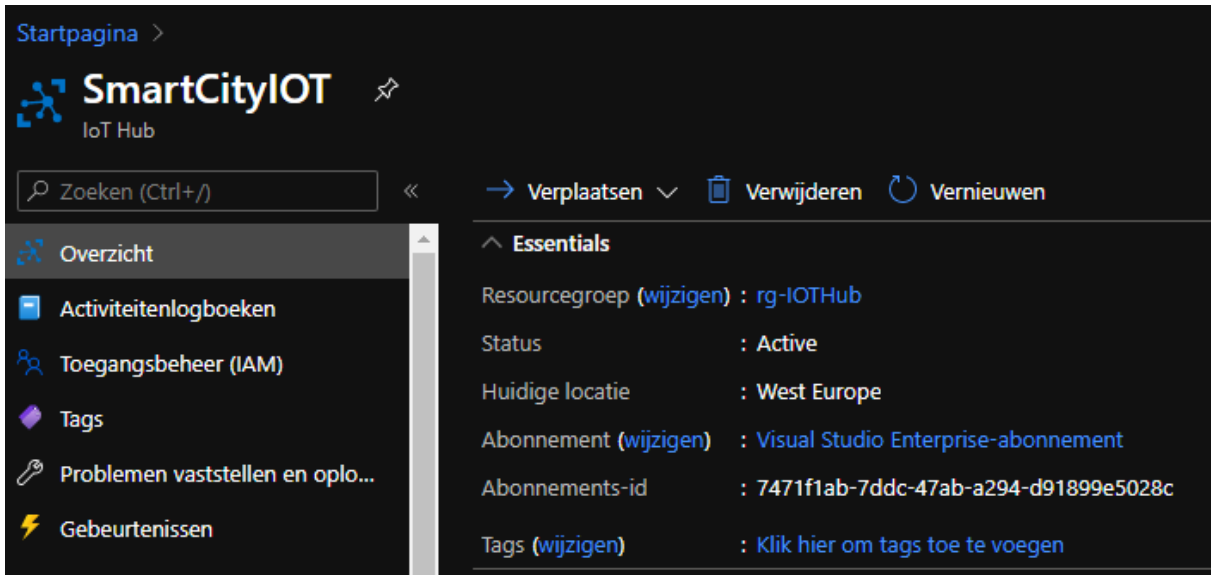
Resultaat

Live aanpassingen, als het device nieuwe data stuurt, wordt dit via de eventhub automatisch aan de SQL DB toegevoegd.

ID	tijd	afstand	beweging	gewicht	leegemaakt	vuur	longitude	latitude	sluikstorten	aantal_sluike...
jq3twyzk5g	1-2-2021 11:51:33	70	1	2964	1	1	4.832242	51.171092	0	6
jq3twyzk5g	1-2-2021 11:50:18	73	0	3914	0	1	4.832242	51.171092	0	5
jq3twyzk5g	1-2-2021 11:49:03	47	0	3266	1	1	4.832242	51.171092	1	5
jq3twyzk5g	1-2-2021 11:47:48	30	1	1694	0	0	4.832242	51.171092	1	7
jq3twyzk5g	1-2-2021 11:46:33	42	0	1181	0	0	4.832242	51.171092	1	6
jq3twyzk5g	1-2-2021 11:45:18	40	1	1577	0	0	4.832242	51.171092	1	3
jq3twyzk5g	1-2-2021 11:44:03	45	1	3426	0	1	4.832242	51.171092	0	5
jq3twyzk5g	1-2-2021 11:42:48	79	0	3371	0	0	4.832242	51.171092	0	10
jq3twyzk5g	1-2-2021 11:41:33	92	1	3407	1	0	4.832242	51.171092	1	7

Data Kiosk

IOT hub



Startpagina >

SmartCityIoT
IoT Hub

Zoeken (Ctrl+/) << → Verplaatsen Verwijderen Vernieuwen

Overzicht

- Activiteitenlogboeken
- Toegangsbeheer (IAM)
- Tags
- Problemen vaststellen en oplo...
- Gebeurtenissen

Essentials

Resourcegroep (wijzigen) : rg-IOTHub

Status : Active

Huidige locatie : West Europe

Abonnement (wijzigen) : Visual Studio Enterprise-abonnement

Abonnements-id : 7471f1ab-7ddc-47ab-a294-d91899e5028c

Tags (wijzigen) : Klik hier om tags toe te voegen

Prijs: F1 – Gratis

Device connecteren via connectiestring

Apparaat-id	Status	Laatste statusupdate (...)	Verificatietype	Aantal cloud-naar-apparaat-berichten
kiosk	Enabled	--	Sas	0

Stream Analytics taak

Nieuwe Stream Analytics taak aanmaken. Invoer is SmartCityIoT (IOT hub), uitvoer is DataKiosk (tabel in SQL database).

Query:

```
SELECT
  IoTHub.EnqueueTime AS tijd,
  temperature AS temperatuur,
  humidity AS vochtigheid,
  aantal_mensen AS aantal_mensen

INTO
  [DataKiosk]

FROM
  [SmartCityIoT]
```

Twilio voor Azure (SMS berichten)

Eerst moeten we een account aanmaken op <https://www.twilio.com/>. Hier is een gratis krediet van ongeveer 15 USD beschikbaar voor hun producten uit te proberen.

Topologie:

Bestaande event hub (gatewaysql) -> nieuwe Stream analytics taak -> nieuwe event hub

Nieuwe event hub maken

Startpagina > SmartCityHerentals >

Event Hub maken

Event Hubs

Naam * ⓘ

notification ✓

Aantal partities ⓘ

2

Bewaarperiode van bericht ⓘ

1

Vastleggen ⓘ

Aan Uit

Nieuwe Stream Analytics taak

Hiermee maakt u een nieuwe Stream Analytics-taak. Kosten worden berekend op basis van de gebruikte resources.

Nieuwe Stream Analytics-taak

Taaknaam *

trigger ✓

Abonnement *

Visual Studio Enterprise-abonnement

Resourcegroep *

rg-SmartCityEventHub

Nieuw

Locatie *

West-Europa

Hostingomgeving ⓘ

Cloud Edge

Streaming-eenheden (1 tot 192) ⓘ

1

☐ Beveilig alle assets met privégegevens die voor deze taak zijn vereist in mijn opslagaccount. ⓘ

Invoer: gatewaysql event hub

Uitvoer: notification event hub

Query:

```
SELECT
    deviceId as ID,
    DATEADD(hh, 1, enqueuedTime) as tijd,
    telemetry.afstand AS afstand,
    telemetry.beweging AS beweging,
    telemetry.gewicht AS gewicht,
    telemetry.leeggemaakt AS leeggemaakt,
    telemetry.sluikestorten AS sluikestorten,
    telemetry.vuur AS vuur,
    CASE
        WHEN deviceId like 'jq3twyzk5g'
            THEN 'Herenthoutseweg 101-87, 2200 Herentals'
            ELSE 'Augustijnenlaan 30, 2200 Herentals'
    END as adres
INTO
    [notification]
FROM
    [fromgatewaysql]
WHERE
    telemetry.vuur = 1
```

Deze query gaat events die in de “gatewaysql” event hub binnen komen, doorsturen naar de nieuwe eventhub “notification”, wanneer de waarde van vuur op 1 staat.


Functie app aanmaken

Vervolgens moeten we een .NET functie app aanmaken.

Een functie-app maken

Basisinformatie Hosten Controleren Tags Beoordelen en maken

Overzicht

 **Functie-app**
door Microsoft

Details

Abonnement	7471f1ab-7ddc-47ab-a294-d91899e5028c
Resourcegroep	rg-Messages
Naam	vuur-sluikstorten
Runtimestack	.NET Core 3.1

Hosten

Opslag (nieuw)

Opslagaccount	storageaccountrgmesb2ce
---------------	-------------------------

Plan (nieuw)

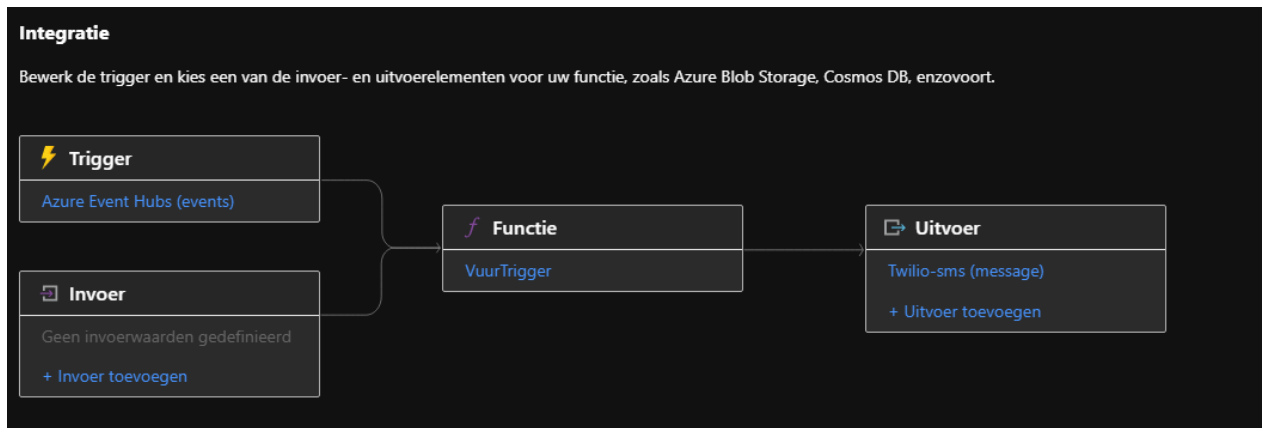
Plantype	Verbruik (serverloos)
Naam	ASP-rgMessages-8479
Besturingssysteem	Windows
Regio	West Europe
SKU	Dynamic

Controleren (nieuw)

Application Insights	Ingeschakeld
Naam	vuur-sluikstorten
Regio	West Europe

(Onze momentele functie app heeft de naam vuur i.p.v. vuur-sluikstorten)

In deze functie app moeten we een functie aanmaken, “VuurTrigger”.



Iedere keer als er een event binnen komt op onze “notification” event hub zal de functie “VuurTrigger” uitgevoerd worden, en een sms gestuurd worden naar de Twilio instellingen.

Trigger

De instellingen van de trigger:

The screenshot shows the 'Trigger bewerken' form with the following settings:

- Bindingstype:** Azure Event Hubs
- Naam van de gebeurtenisparameter:** events
- Event Hub-verbinding:** SmartCityHerentals_PreviewDataPoli...
- Event Hub-naam:** notification
- Event Hub-consumentengroep:** \$Default
- Kardinaliteit van de Event Hub:** Many
- Data type of the trigger input. Choose ...:** (Dropdown menu)

De event hub verbinding is een verbinding met een policy die read rechten heeft op de notification event hub.

Uitvoer

Instellingen voor de uitvoer:

Uitvoer bewerken
✕

Opslaan
✕ Negeren
...

Bindingstype

Twilio-sms

Naam van de berichtparameter * ⓘ

message

Instelling voor de account-SID * ⓘ

TwilioAccountSid

Instelling voor het verificatietoken * ⓘ

TwilioAuthToken

Vanaf telefoonnummer ⓘ

+16515386964

Berichttekst ⓘ

Telefoonnummer is het nummer dat je kan verkrijgen op Twilio: <https://www.twilio.com/console>

Account-SID en verificatietoken zijn de namen van de omgevingsvariabelen die nog geconfigureerd moeten worden. Deze configureer je in de functie-app/Instellingen/Configuratie. De waardes van deze variabelen zijn ook terug te vinden op de Twilio console.

TwilioAccountSid	Verborgen waarde. Klik om de waarde weer te geven	App-configuratie
TwilioAuthToken	Verborgen waarde. Klik om de waarde weer te geven	App-configuratie

Funcitie code

De vorige instellingen maakten normaal al de code in “function.json” aan. Dit bestand ziet er als volgt uit:

```
{
  "bindings": [
    {
      "type": "eventHubTrigger",
      "name": "events",
      "direction": "in",
      "eventHubName": "notification",
      "cardinality": "many",
      "connection": "SmartCityHerentals_PreviewDataPolicy_EVENTHUB",
      "consumerGroup": "$Default"
    },
    {
      "name": "message",
      "accountSidSetting": "TwilioAccountSid",
      "authTokenSetting": "TwilioAuthToken",
      "from": "+16515386964",
      "direction": "out",
    }
  ]
}
```

```
"type": "twilioSms"
}
]
}
```

Dit is een “in” binding om de events te ontvangen, met de naam events. En een “out” binding om via Twilio berichten te sturen, met de naam message. Deze namen hebben we later nodig in onze functie code.

Om correct met de Twilio API te kunnen werken moet eerst een bestand “function.proj” worden toegevoegd. Hierin moeten we de SDK en versie definiëren.

Code:

```
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <TargetFramework>netstandard2.0</TargetFramework>
  </PropertyGroup>
  <ItemGroup>
    <PackageReference Include="twilio" Version="5.27.2"/>
  </ItemGroup>
</Project>
```

De code van onze functie, dus run.csx is als volgt:

```
#r "Newtonsoft.Json"
#r "Twilio"
#r "Microsoft.Azure.WebJobs.Extensions.Twilio"

using System;
using Microsoft.Extensions.Logging;
using Newtonsoft.Json;
using Microsoft.Azure.WebJobs.Extensions.Twilio;
using Twilio.Rest.Api.V2010.Account;
using Twilio.Types;

public static async Task Run(string events, IAsyncCollector<CreateMessageOptions> message, ILogger log)
{
    log.LogInformation($"C# Queue trigger function processed: {events}");

    // Bericht samenstellen text + ID
    dynamic order = JsonConvert.DeserializeObject(events);
    string msg = "Er is vuur gedetecteerd in de vuilbak met ID " + order.ID + ". Deze vuilbak bevindt zich op " + order.adres + ".";

    // To phone number instellen
    CreateMessageOptions smsText = new CreateMessageOptions(new PhoneNumber("+32471212511"));

    // Bericht als body doorsturen
    smsText.Body = msg;
```

```
await message.AddAsync(smsText);  
}
```

Deze code stuurt voor iedere event die in de notification event hub binnen komt een bericht naar de gsm met nummer +32471212511, met bericht "Er is vuur gedetecteerd in de vuilbak met ID *ID*. Deze vuilbak bevindt zich op *adres*".

Resultaat

Er wordt vuur gedetecteerd en de waardes worden doorgestuurd naar de database.

ID	tijd	afstand	beweging	gewicht	leegemaakt	vuur	longitude	latitude	sluikstorten
jq3twyzk5g	27-1-2021 15:15:18	83	1	2403	1	1	4.832242	51.171092	1

Het bericht wordt verstuurd.



Azure Defender (security)

Azure Defender is een security tool op het Microsoft Azure platform. Deze tool voorziet beveiliging voor virtuele machines, databases, web applicaties, en meer. Ook kunnen er security alerts geconfigureerd worden.

Voor onze omgeving kunnen wij gebruik maken van Azure Defender for SQL voor de database, en Azure Defender for key vault voor onze encryptiesleutels.

Azure Defender for SQL

Azure Defender for SQL biedt de volgende voordelen aan:

- Vulnerability assessment
Dit is een scanning service die het mogelijk maakt om potentiële database vulnerabilities te detecteren, tracken, en helpt om deze vulnerabilities te remediëren. Assessment scans geven een overzicht van de security staat van de SQL database.
- Advanced threat protection
Deze detectie service monitort continu de SQL database voor bedreigingen zoals SQL injectie, bruteforce attacks, en misbruik van privilege. De service biedt actie-georiënteerde security alerts aan in Azure Security Center met details van verdachte activiteit, en begeleiding voor de bedreigingen te elimineren.

Enkele voorbeelden van alerts die Azure Defender for SQL voorziet:

- Potentiële SQL injectie aanvallen
Dit wordt gedetecteerd wanneer applicaties een foutief SQL statement genereren in de database.
- Abnormale database toegang en query patronen
Bijvoorbeeld, een abnormaal aantal gefaalde inlogpogingen met verschillende credentials (een brute force poging)
- Verdachte database activiteit
Bijvoorbeeld, een legitieme gebruiker die met een SQL Server verbindt vanaf een gebreachte computer die gecommuniceerd heeft met een crypto-mining C&C server.

Deze alerts bevatten details van het incident dat de alert triggerde, inclusief recommandaties voor hoe het probleem te investigeren en op te lossen.

Azure Defender for Key Vault

Azure Defender for Key Vault is een cloud service die encryptie keys en andere secrets, zoals certificaten, connectie strings, en wachtwoorden beveiligd.

Wanneer er abnormale activiteiten zich voordoen dan genereert Azure Defender for Key Vault alerts, met een optie om een mail te sturen naar leden van de organisatie.

Deze alerts bevatten details van het incident dat de alert triggerde, inclusief recommandaties voor hoe het probleem te investigeren en op te lossen.

Implementatie

Azure Defender biedt een proefversie aan voor 30 dagen om de verschillende services uit te proberen. Hierna zijn er vaste prijzen per resource per maand. De prijzen voor onze resources zijn als volgt:

15 USD per maand voor de SQL DB + 0.02 USD voor ons opslagaccount + 0.02 USD voor onze sleutelkluis

= 15.04 USD per maand

De proefperiode loopt tot 24 februari, ons project is afgelopen 18 februari. Er zullen dus geen kosten zijn voor Azure Defender in onze conceptfase.

Installeren agents

Optimaal profiteren van Security Center door agents voor het verzamelen van gegevens in te schakelen

Als u beveiligingswaarschuwingen en aanbevelingen wilt ontvangen, moeten er voor het verzamelen van gegevens agents op uw virtuele machines zijn geïnstalleerd.

[Meer informatie >](#)

Agents automatisch installeren

De Log Analytics-agent wordt automatisch op alle virtuele machines in het geselecteerde abonnement geïnstalleerd.

^ Selecteer abonnementen waarvoor agents worden geïnstalleerd 0 beheerde resources

Naam	Niet-beveiligde ...
Visual Studio Enterprise-abonnement	0

[Installeer agents](#)

Als de SCOM- of OMS-agent al lokaal is geïnstalleerd op een VM, wordt de extensie van de Log Analytics-agent nog steeds geïnstalleerd en verbonden met de geconfigureerde werkuimte.

Agents handmatig installeren

Als u al een andere werkuimte hebt, kunt u er virtuele machines mee verbinden en zelf agents installeren vanuit de [De pagina Prijzen en instellingen](#)

Doorgaan zonder agents te installeren

Veel belangrijke beveiligingsfuncties werken niet als u geen agents installeert.

[Doorgaan zonder agents te installeren](#)

Agents automatisch installeren

De Log Analytics-agent wordt automatisch op alle virtuele machines in het geselecteerde abonnement geïnstalleerd.

^ **Alles is ingesteld. Voor al uw Azure-abonnementen is de automatische installatie van de agent ingeschakeld**

Microsoft Azure

Resources, services en documenten zoeken (G+T)

SenVanBroekhoven@st... STUDENT AMBASSADORS

Startpagina >

Beveiligingscentrum | Overzicht

Het abonnement Visual Studio Enterprise-abonnement wordt weergegeven.

Zoeken (Ctrl+F)

Abonnementen Nieuw

Algemeen

- Overzicht
- Aan de slag
- Aanbevelingen
- Beveiligingswaarschuwingen
- Inventaris
- Community

Cloud Security

- Beveiligingsscore
- Naleving van regelgeving
- Azure Defender

Beheer

- Prijzen en instellingen
- Beveiligingsbeleid
- Beveiligingsoplossingen
- Werkstroomautomatisering
- Dekking
- Cloudconnectors (preview)

1 Azure-abonnementen 17 Actieve aanbevelingen -- Beveiligingswaarschuwingen

Beveiligingsscore

Huidige veiligheidsscore

24% 5/8 punten

VOLTOOID Besturingselementen 2/8

VOLTOOID Aanbevelingen 4/1

[Uw veiligheidsscore verbeteren >](#)

Naleving van regelgeving

Geen nalevingscontrole

[De naleving verbeteren >](#)

Azure Defender

Resource dekking 100% Voor volledige beveiliging schakelt u 0 resourceplannen in

Waarschuwingen op ernstniveau

Geen beveiligingswaarschuwingen

[Uw mogelijkheden voor bescherming tegen bedreigingen verbeteren >](#)

Inventaris

Onbewaakte virtuele machines 0 All VMs are monitored

Totaal aantal resources 9

Niet in orde (0) In orde (1) Niet van toepassing (0)

[Uw resources verkennen >](#)

Inzichten

Meestvoorkomende aanbevelingen (op resources)

- Diagnostische logboeken in Azure Strea... 2
- Azure Defender for App Service moet zij... 1
- Storage accounts should restrict network... 1
- Openbare toegang tot een opslagaccou... 1

Besturingselementen met de hoogst mogelijke verhoging

- Remediate vulnerabilities +26% (6p)
- Manage access and permissions +17% (4p)
- Encrypt data in transit +17% (4p)

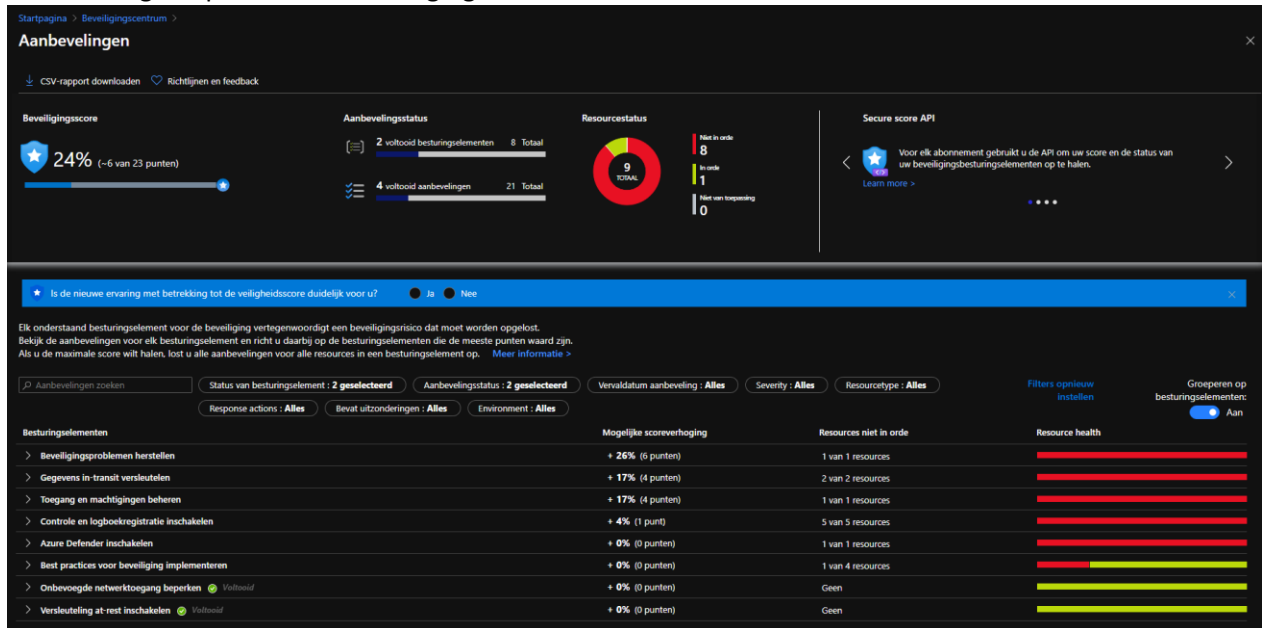
[Besturingselementen weergeven >](#)

Azure Security Center-community

Neem deel aan de Azure Security Center-community op GitHub om te communiceren met andere klanten en experts en om te leren, feedback te geven en kennis te delen over Security Center.

[Azure-community weergeven >](#)

Aanbevelingen oplossen om beveiligingsscore te verbeteren



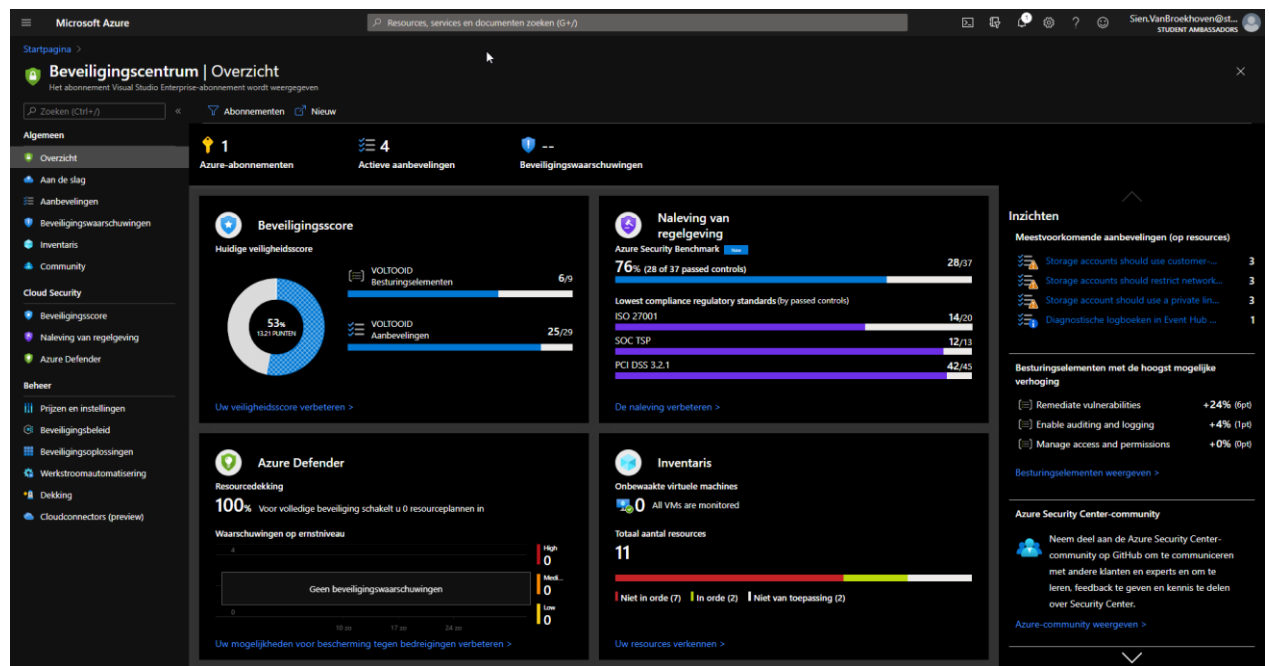
Wachten tot Azure Defender implementatie voltooid is.

Volgende beveiligingsstappen uitvoeren:

1. SQL Database toevoegen aan Azure Defender
2. Openbare toegang tot opslagaccounts blokkeren
3. Beveiligde overdracht naar opslagaccounts inschakelen
4. HTTPS vereisen bij functie-app
5. Azure Defender voor Azure SQL Database Servers inschakelen
6. Auditing inschakelen voor SQL DB
7. Basislijn firewall instellen
8. Basislijn geauthentiseerde users DB instellen

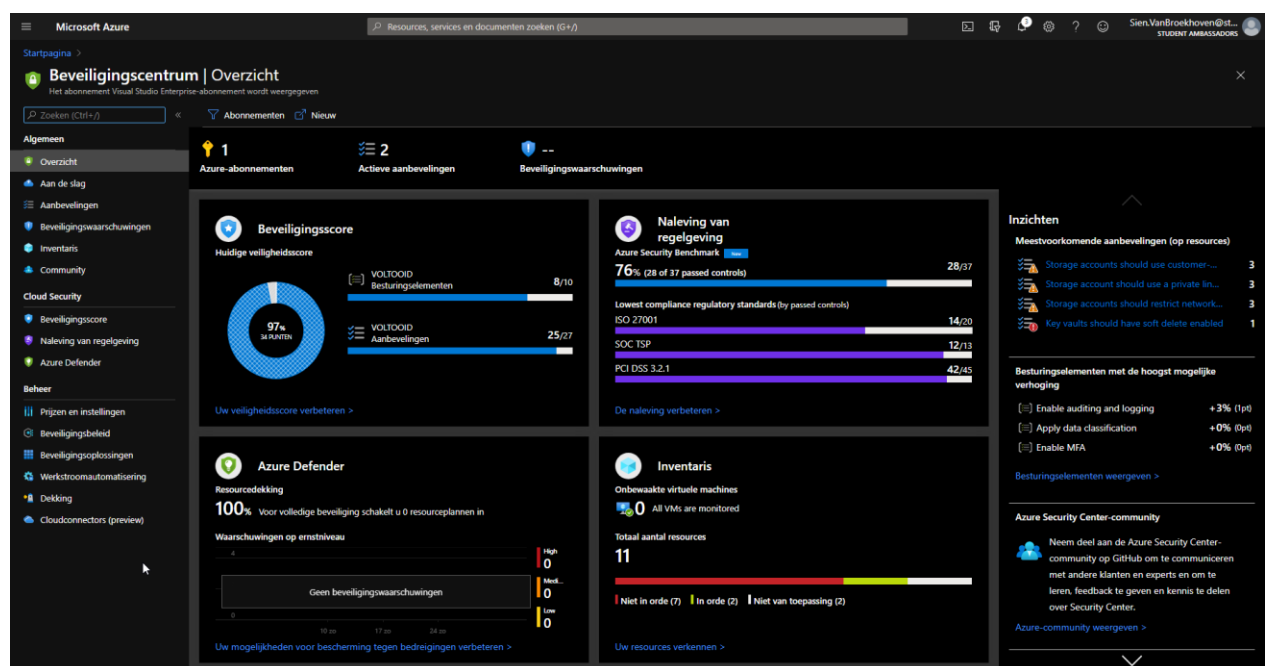
Inventaris beveiligen:

1. Contact email voor security issues instellen op subscriptie
2. Firewall instellen voor storage accounts

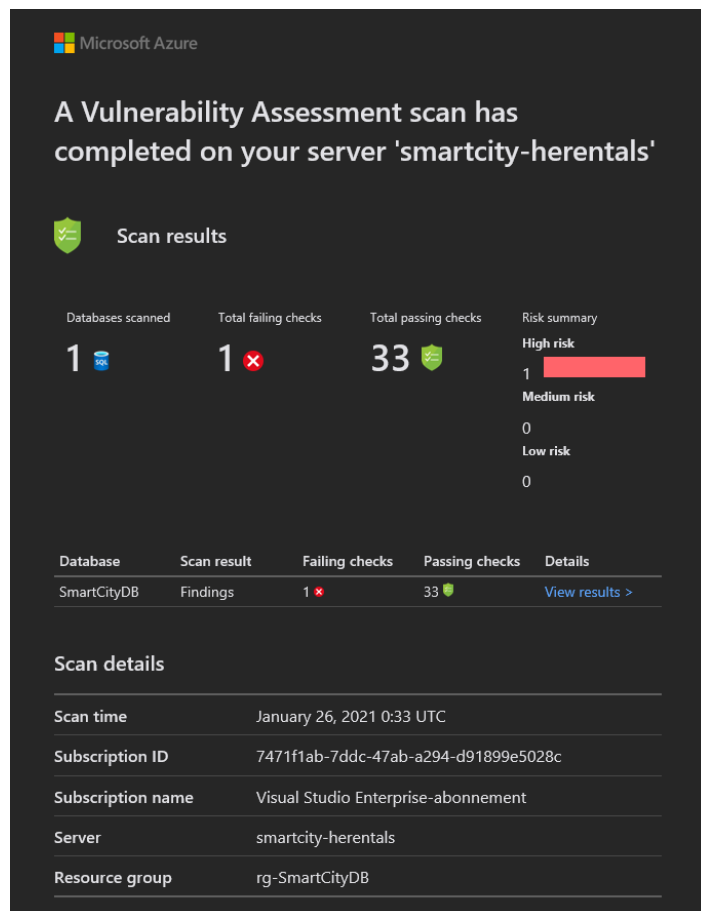


Sommige veranderingen zijn nog niet doorgevoerd, kan een tijdje duren.

Volgende dag:



Wanneer er een vulnerability scan op onze SQL database uitgevoerd wordt, dan wordt er ook een mail met de resultaten gestuurd.

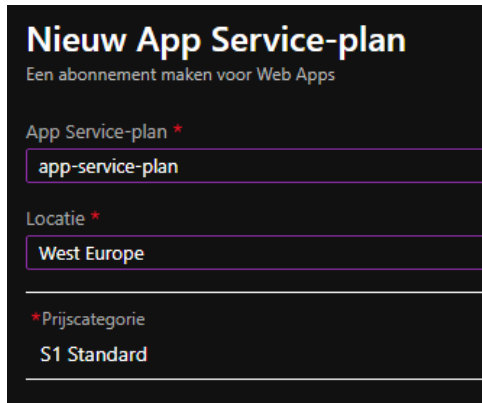


Hosten Laravel backend op Azure

<https://medium.com/swlh/deploy-laravel-5-8-on-azure-web-app-2019-3514eb2fd1af>

Web-app + MySQL

Nieuwe Web-app + MySQL aanmaken.



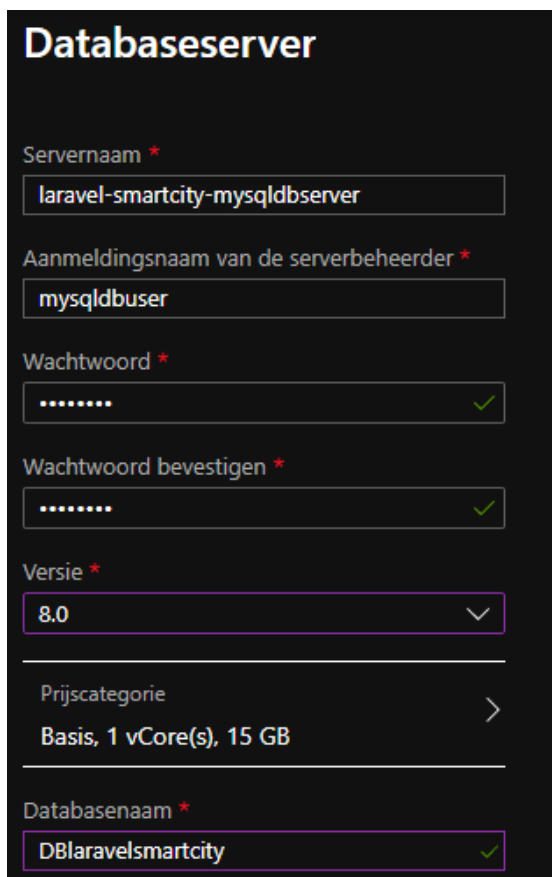
Nieuw App Service-plan
Een abonnement maken voor Web Apps

App Service-plan *
app-service-plan

Locatie *
West Europe

* Prijscategorie
S1 Standard

Prijs = ~ 61 euro per maand (pay as you go), wordt bepaald op aantal instances en de grootte. Kan nog aangepast worden.



Databaseserver

Servernaam *
laravel-smartcity-mysqldbserver

Aanmeldingsnaam van de serverbeheerder *
mysqlbuser

Wachtwoord *
..... ✓

Wachtwoord bevestigen *
..... ✓

Versie *
8.0 ✓

Prijscategorie
Basis, 1 vCore(s), 15 GB >

Databasenaam *
DBlaravelsmartcity ✓

Prijs Databaseserver ~ 25 euro per maand

Instellingen aanpassen naar PHP in configuration.

Vernieuwen
Opslaan
Negeren

Toepassingsinstellingen
Algemene instellingen *
Standaarddocumenten
Pa...

Stack-instellingen

Stack

PHP

PHP-versie

PHP 7.3

Composer extensie toevoegen

[Startpagina](#) >
[laravel-smartcity](#) >

Extensie toevoegen

*Uitbreiding kiezen

Composer

*Juridische voorwaarden

Accepted

Verbinding maken met GitHub waar onze Laravel configuratie staat via implementatie/implementatiecentrum. Wanneer er naar de GitHub gepusht wordt, wordt de Azure configuratie ook aangepast.

Bladeren Vernieuwen Verbinding verbreken Synchroniseren Referenties implementatie				
Bron	Opslagplaats			
GitHub	https://github.com/UpSmartOrg/UpSmartAppApi			
Build	Vertakking	Type broncodebeheer		
Kudu	master	Git		
TJD	STATUS	COMMIT-ID (AUTEUR)	CHECK-INBERICHT	LOGBOEKEN
Monday, February 15, 2021				
9:45:43 AM GMT+1	Running deployment command...	696f222 (Seppe Alaerts)	Merge pull request #19 from JasperGuldenbops/sa-quick-survey Sa quick survey	

.env kopiëren en artisan key generaten.

```

      _ _ _ _ _
     / \ / \ / \ / \
    _ _ / \ / \ / \ / \
   ( _ / \ / \ / \ / \ )
     ( _ _ _ _ ) ( _ _ _ )
      ( _ _ _ _ ) ( _ _ _ )

U beheert uw web-appomgeving door algemene opdrachten uit te voeren ('mkdi

C:\home\site\wwwroot>copy .env.example .env
1 file(s) copied.

C:\home\site\wwwroot>php artisan key:generate
Application key set successfully.

C:\home\site\wwwroot>_
```

.env aanpassen naar correcte databasegegevens en php artisan migrate uitvoeren.

```

      _ _ _ _ _
     / \ / \ / \ / \
    _ _ / \ / \ / \ / \
   ( _ / \ / \ / \ / \ )
     ( _ _ _ _ ) ( _ _ _ )
      ( _ _ _ _ ) ( _ _ _ )

U beheert uw web-appomgeving door algemene opdrachten uit te voeren ('mkdir',

C:\home\site\wwwroot>php artisan migrate
Migration table created successfully.
Migrating: 2019_12_14_000001_create_personal_access_tokens_table
Migrated: 2019_12_14_000001_create_personal_access_tokens_table (0.9 seconds)
Migrating: 2021_01_19_000000_create_failed_jobs_table
Migrated: 2021_01_19_000000_create_failed_jobs_table (0.24 seconds)
Migrating: 2021_01_19_000001_create_roles_table
Migrated: 2021_01_19_000001_create_roles_table (0.39 seconds)
Migrating: 2021_01_19_000002_create_users_table
Migrated: 2021_01_19_000002_create_users_table (0.98 seconds)
Migrating: 2021_01_19_000003_create_user_roles_table
Migrated: 2021_01_19_000003_create_user_roles_table (1.41 seconds)
Migrating: 2021_01_19_000004_create_password_resets_table
Migrated: 2021_01_19_000004_create_password_resets_table (0.58 seconds)
Migrating: 2021_01_19_000100_create_information_table
Migrated: 2021_01_19_000100_create_information_table (0.89 seconds)
Migrating: 2021_01_20_001000_create_surveys_table
Migrated: 2021_01_20_001000_create_surveys_table (0.81 seconds)
Migrating: 2021_01_20_001100_create_open_questions_table
Migrated: 2021_01_20_001100_create_open_questions_table (0.78 seconds)
Migrating: 2021_01_20_001300_create_multiplechoice_questions_table
Migrated: 2021_01_20_001300_create_multiplechoice_questions_table (0.87 seconds)
Migrating: 2021_01_20_001301_create_multiplechoice_items_table
Migrated: 2021_01_20_001301_create_multiplechoice_items_table (0.81 seconds)
Migrating: 2021_01_20_002001_create_answers_table
Migrated: 2021_01_20_002001_create_answers_table (1.97 seconds)
Migrating: 2021_02_02_085057_create_zones_table
Migrated: 2021_02_02_085057_create_zones_table (0.28 seconds)
Migrating: 2021_02_02_085711_create_bin_info_table
Migrated: 2021_02_02_085711_create_bin_info_table (1.43 seconds)

C:\home\site\wwwroot>_
```

Web.config toevoegen

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
```

```

<system.webServer>

  <urlCompression doDynamicCompression="true" doStaticCompression="true"
dynamicCompressionBeforeCache="true"/>

  <staticContent>
    <remove fileExtension=".svg" />
    <mimeTypeMap fileExtension=".svg" mimeType="image/svg+xml" />
    <mimeTypeMap fileExtension=".woff" mimeType="application/font-woff" />
    <clientCache httpExpires="Sun, 29 Mar 2022 00:00:00 GMT" cacheControlMode="UseExpires" />
    <remove fileExtension=".woff" />
    <mimeTypeMap fileExtension=".woff" mimeType="application/font-woff" />
    <remove fileExtension=".woff2" />
    <mimeTypeMap fileExtension=".woff2" mimeType="font/x-woff" />
  </staticContent>

  <httpProtocol>
    <customHeaders>
      <add name="Strict-Transport-Security" value="max-age=31536000; includeSubDomains"/>
      <add name="Access-Control-Allow-Headers" value="X-Requested-With,Content-Type" />
      <add name="Access-Control-Allow-Methods" value="POST,GET,OPTIONS,DELETE,PUT,PATCH" />
    </customHeaders>
  </httpProtocol>

  <rewrite>
    <rules>

      <rule name="Laravel" stopProcessing="true">
        <match url="^" ignoreCase="false" />

        <conditions logicalGrouping="MatchAll">
          <add input="{REQUEST_FILENAME}" matchType="IsDirectory" negate="true" />
          <add input="{REQUEST_FILENAME}" matchType="IsFile" negate="true" />
        </conditions>

        <action type="Rewrite" url="index.php" appendQueryString="true" />

      </rule>

    </rules>
  </rewrite>


  <handlers>
    <remove name="OPTIONSVerbHandler" />
    <remove name="PHP56_via_FastCGI" />
    <add name="PHP56_via_FastCGI" path="*.php" verb="GET,HEAD,POST,PUT,DELETE,OPTIONS"
modules="FastCgiModule" scriptProcessor="D:\Program Files (x86)\PHP\v7.3\php-cgi.exe"
resourceType="Either" />
  </handlers>

```



```
<customErrors mode="Off" />
<httpErrors errorMode="Detailed"></httpErrors>
</system.webServer>
</configuration>
```

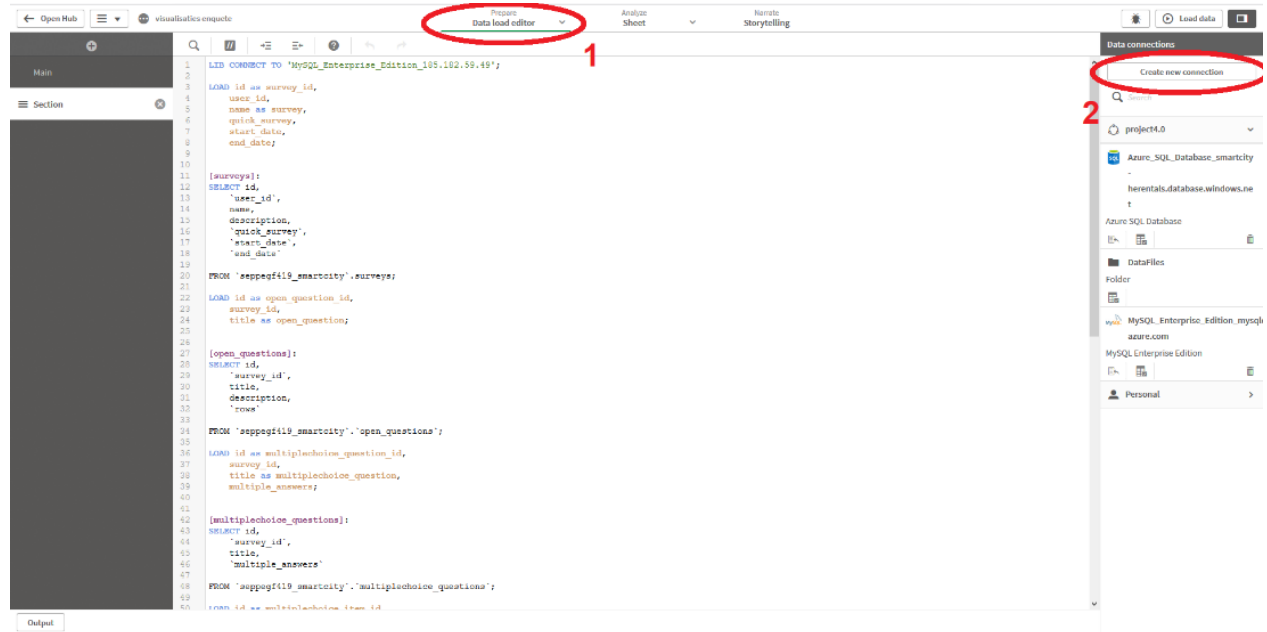
Root pad wijzigen:

Virtuele toepassingen en mappen		
 Nieuwe virtuele toepassing of map		
Virtueel pad	Fysiek pad	Type
/	site\wwwroot\public_html	Toepassing

BI dashboards

Script

Het script voor de bi dashboards kan u terugvinden in de qvf file samen met de dashboards zelf. Als u de qvf file wilt inladen zal u errors krijgen. Het script linkt namelijk naar onze database. Je zal dus eerst met Qlik een connectie moeten maken naar een andere database met dezelfde structuur. Dit doe je in het tablad data load editor.

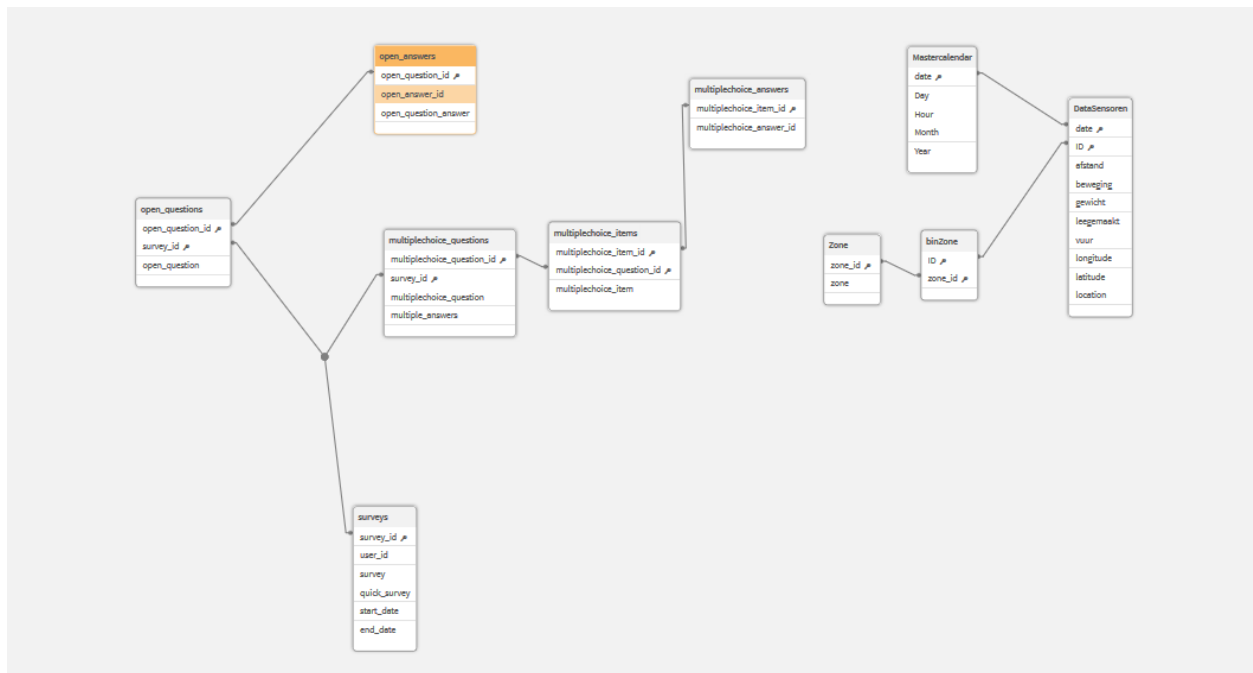


Houd er rekening mee dat de from statements en de lib connect to statement anders kunnen zijn.

Eens je deze aangepast hebt zal je script normaal zonder errors inladen.

Datamodel

Op de foto hieronder kan je ons datamodel van Qlik terugvinden. Dit is het datamodel dat je zou moeten verkrijgen wanneer je het script juist hebt ingeladen. Merk op dat we eigenlijk 2 verschillende datamodellen hebben. Aan de linkerkant het datamodel van de enquêtes en rechts dat van de vuilbakken.



Visualisaties

De meeste visualisaties zijn vrij voor de hand liggend. Je kan de code dan ook volledig terugvinden in de qvf file. Toch gaan we sommige visualisaties hier wat meer toelichten om te bekijken wat er precies gebeurt.

KPI voorspelling

We beginnen met de voorspelling die we gemaakt hebben. Dit is een kpi die gaat kijken wanneer de vuilbak vol gaat zijn. Dit doen we doormiddel van volgende code:

```
(100/(((170 - FirstSortedValue(afstand, -date))/ 170)*100/((Date(HuidigeDate)-
Date(LaatsteLeging)))))-(Date(HuidigeDate)- Date(LaatsteLeging))
```

We kunnen dit opdelen in verschillende delen:

```
(100/(((170 - FirstSortedValue(afstand, -date))/ 170)*100/((Date(HuidigeDate)-
Date(LaatsteLeging))))))
```

We gaan berekenen op basis van de vullingsgraad en het aantal dagen sinds de laatste leging die we nu hebben hoeveel dagen het duurt om van 0% tot 100% te gaan.

```
(Date(HuidigeDate)- Date(LaatsteLeging))
```

We gaan kijken hoeveel dagen er tussen vandaag en de laatste leging zitten.

Als we beide weten trekken we deze van elkaar af dus bv. 16 dagen om van 0-100% te gaan en 7 dagen sinds laatste leging wil zeggen dat de vuilbak vol zit binnen 9dagen.

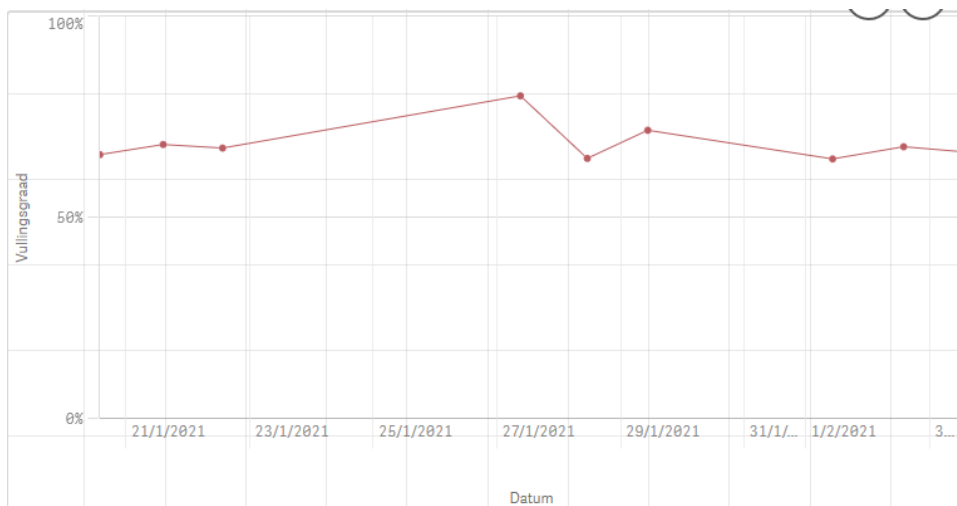
Voorspelling 100% vol

9 dagen

We zien dat hier gebruik wordt gemaakt van de functie `FirstSortedValue()`. Deze functie vind je een paar keer terug in de bijgevoegde qvf file en zorgt er voor dat we alleen de laatste value gaan gebruiken op basis van een andere value. Bijvoorbeeld: `FirstSortedValue(afstand, -date)` wil zeggen dat we de afstand gaan gebruiken die het laatst is binnengekomen.

Linechart vullingsgraad per dag - datum

De andere visualisatie die extra toelichting nodig heeft is de volgende:



Voor deze visualisatie moesten we de vullingsgraad per dag pakken en dus niet per interval (1 uur). Hiervoor moesten we dus bij vullingsgraad de volgende code toepassen:

```
max(aggr(((170 - avg(afstand))/ 170),Day))
```

We zien hier dus dat we de max pakken van het gemiddelde van de vullingsgraad per dag. We gebruiken `aggr` voor meer gevorderde aggregaties. De code hierboven gaat dus de eerste value van een dag pakken en deze steeds doorgeven. Zo krijgen we een grafiek die smoother is. Ook hebben we de null values genegeerd aangezien onze vuilbak niet dag en nacht op stond.

Tooltip laatst leeggemaakt

Om te kijken wanneer de vuilbak het laatst is leeggemaakt gebruiken we volgende code.

```
Date(max({<leeggemaakt={1}>} date))
```

In onze data krijgen we een 1 aan als de vuilbak is leeggemaakt en een 0 indien dit niet is gebeurd. In de volgende code gaan we de laatste datum (max date) opvragen wanneer leeggemaakt gelijk was aan 1. Dit geeft een getal terug dus gebruiken we de Date() functie om dit om te zetten naar een datum.

Als je de qvf file gaat inladen in Qlik zal je alle visualisaties en bijhorende code te zien krijgen.

Embedding

Om te beginnen met de embedding moeten we eerst verbinding maken met Qlik. Dit doen we als volgt:

```
1 connect();
2
3 //connecteren met qlik
4
5 async function connect() {
6   const urlQlikServer = "https://r0747459.eu.qlikcloud.com";
7   const urlLoggedin = "/api/v1/audits"; //kijken of je ingelogd bent ben GET request
8   const urlLogin = "/login";
9   const webIntegrationId = 'UoY07E8550fz12d6mGO-KV3znDzOMCjm'; //kan gevonden worden in qlik bij management console/webintegration
10
11   //nakijken of er ingelogd is
12   return await fetch(`${urlQlikServer}${urlLoggedin}`, {
13     credentials: 'include',
14     headers: {
15       'Qlik-Web-Integration-ID': webIntegrationId
16     }
17   });
18 }
19 .then(async function(response)
20 {
21   //als user niet ingelogd is redirecten naar inlog pagina
22   if(response.status===401){
23     const url = new URL(`${urlQlikServer}/login`);
24     url.searchParams.append('returnto', 'https://testyentl.sinnners.be/'); //return to je website
25     url.searchParams.append('qlik-web-integration-id', webIntegrationId);
26     window.location.href = url;
27   }
28 }
29 )
30 .catch(function(error)
31 {
32   console.error(error);
33 });
34
35
36 }
```

Nu Qlik weet wie we zijn moeten we een configuratie maken voor de embedding te gaan doen.

```
//config maken voor embedding
var config1 = {
  host: "r0747459.eu.qlikcloud.com", //adres van qlik server
  prefix: "/", //alleen aanpassen bij virtual proxy
  port: 443, //default poort, aanpassen indien gebruik andere poort
  isSecure: true, //true bij https, false bij http
  webIntegrationId: 'UoY07E8550fz12d6mGO-KV3znDzOMCjm' //kan gevonden worden in qlik bij management console/webintegration
};

require.config( {
  baseUrl: (config1.isSecure ? "https://" : "http://" ) + config1.host + (config1.port ? ":" + config1.port : "") + config1.prefix + "resources",
  webIntegrationId: config1.webIntegrationId
} );
```

Qlik kan zelf ook errors geven. Deze code zit ook in het javascript van de embedding.

```
//qlik errors

require( ["js/qlik"], function ( qlik ) {

    qlik.on( "error", function ( error ) {
        $( '#popupText' ).append( error.message + "<br>" );
        $( '#popup' ).fadeIn( 1000 );
    } );
    $( "#closePopup" ).click( function () {
        $( '#popup' ).hide();
    } );
});
```

Als volgende moeten we de applicatie waarvan de visualisaties komen declareren en de visualisaties ophalen. We geven de visualisaties een naam om ze makkelijk te kunnen gebruiken zonder de id te kennen.

```
//open apps

//syntax : var app = qlik.openApp("applicatie id te vinden in de link van je qlik", te_gebruiken_config)

var app = qlik.openApp("b08cba55-8ec6-4c3b-a586-6f6f8be3d789", config1);

//get objects

//syntax : app.getObject('id voor in html', 'object ID in Qlik');

app.getObject('kpiAfstandTotAfval', '28cac925-3de3-4be2-8c5a-f05cb896f384');
app.getObject('passantenChart', 'UNPQ');
app.getObject('afvalChartLine', '005a79fc-9851-492e-a695-c3b63f2d3865',{noSelections: true}); // noSelections:true om geen selecties toe te staan
app.getObject('dataDrilldown', 'e18ea547-3b7d-4e92-8493-1face055384e');
app.getObject('map', 'a2ca22e9-fc27-4dfb-a1ff-40c01873e49b');
app.getObject('kpilaatsteleging', '91cc640a-d2a5-49cb-9409-ab811721b721');
app.getObject('kpiPrediction', 'b410319f-8c4f-4450-8ea6-099b49c56afd');

app.getObject('kpiTotaal', 'ppKugT');
app.getObject('filterSurvey', 'xJRVUJ');
app.getObject('filterQuestions', 'bfpsc');
app.getObject('chartMultiple', 'uXek');
app.getObject('chartOpen', 'jPhDEz');
app.getObject('kpiOpen', 'QREatz');
app.getObject('kpiMultiple', 'YPbuw');
```

Tot slot hebben we nog enkele functies om bijvoorbeeld de data te herladen of de filters weg te doen. Deze functies gaan we dan later toewijzen aan knoppen in onze html.

```
//functies voor buttons

$("#ClearAll").click(function() {
    app.clearAll(); // alle selecties weghalen
});
$("#reload").click(function(){
    app.doReload();
    app.doSave(); // alle data opnieuw inladen
})

});
```

Het javascript is af. Nu gaan we naar de html. In de html beginnen we met linken te leggen naar de css en het javascript gehost door Qlik. Vervolgens leggen we ook een verbinding naar het javascript dat we juist gemaakt hebben.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
  <meta name="description" content="Qlik Sense Embedding Tutorial" />
  <meta name="author" content="Yentl Stroobants" />
  <link rel="icon" type="image/x-icon" href="/assets/img/favicon.ico" />
  <title>Vuilbakken</title>
  <link rel="stylesheet" href="https://r8747459.eu.qlikcloud.com/resources/autogenerated/qlik-styles.css"> <!-- Qlik css linken-->
  <link rel="stylesheet" href="/TM_Tutorial.css">
  <script type="text/javascript" src="https://r8747459.eu.qlikcloud.com/resources/assets/external/requirejs/require.js"></script><!-- Qlik javascript linken-->
  <script src="TM_Tutorial.js"></script> <!-- Embedding javascript linken-->

```

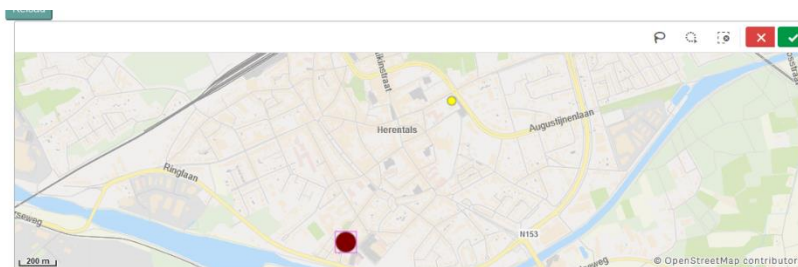
Vervolgens om de visualisaties op te roepen is er niet veel nodig. Je kiest doormiddel van bootstrap hoe de layout moet zijn en je plakt eigenlijk gewoon heel makkelijk de visualisaties erin. Dit doe je door als id de naam in te geven die je in het javascript mee gaf. In dit geval is dat 'map', als class geven we dan qvobject mee. In het voorbeeld staat er nog ml-0 bij om de margin links weg te doen dit is niet nodig en puur voor lay-out. Als je dit allemaal gedaan hebt en gedeployed hebt naar je website kan je de visualisaties bekijken op je website.

```

<div class="container">
  <div class="col-12">
    <div id="map" class="qvobject ml-0"></div>
  </div>
</div>

```

Het resultaat:



Laatste leging
3/2/2021

Voorspelling 100% vol
4 dagen

Afstand tot afval
38 cm

🔍 Date drill down

Jan
Feb

Applicatie

Overzicht

De webapplicatie is de interface tussen de gebruiker en het systeem. De app is ingedeeld in 2 delen. Enerzijds is er de kiosk-app, waar iedereen toegang tot heeft, en anderzijds is er het Smart City platform waar een werknemer van Herentals kan inloggen en informatie kan raadplegen en bewerken.

De kiosk-app wordt getoond op een kiosk die zich bevindt op een openbare plaats in Herentals. Hier kunnen voorbijgangers informatie raadplegen over de stad. Daarnaast kan men een (korte) bevraging invullen, waardoor stad Herentals zijn diensten kan verbeteren en afstemmen op de noden van de inwoners.

Het Smart City platform is de app voor werknemers van stad Herentals. Deze is beschikbaar op de url <http://smartcity.seppealaerts.be>. Om de app te gebruiken, zal een gebruiker moeten inloggen. Op basis van welke rollen er aan de ingelogde gebruiker zijn toegekend, zal de inhoud getoond worden waar de gebruiker toegang tot heeft. Er zijn 4 verschillende rollen:

- De admin. Kan gebruikers van de app raadplegen, toevoegen, bewerken en verwijderen. Heeft toegang tot elk Dashboard.
- De groendienst medewerker. Kan informatie over de slimme vuilnisbakken raadplegen, bewerken en verwijderen. Kan vuilbakken toevoegen aan zones. kan zones raadplegen, toevoegen, bewerken en verwijderen. Heeft enkel toegang tot het Groendienst Dashboard.
- De participatie medewerker. Kan bevestigingen raadplegen, toevoegen, bewerken en verwijderen. Deze bevestigingen worden getoond op de kiosk app, waarop ze ingevuld kunnen worden. Heeft enkel toegang tot het Participatie Dashboard.
- De communicatie medewerker. Kan nieuwsberichten raadplegen, toevoegen, bewerken en verwijderen. Deze berichten worden getoond op de kiosk app. Heeft enkel toegang tot het Communicatie Dashboard.

Een gebruiker kan meerdere rollen toegewezen krijgen, waardoor de gebruiker in kwestie toegang heeft tot meerdere Dashboards.

Daarnaast kunnen er ook externe gebruikers worden toegevoegd. Dit zijn gebruikers die niet in dienst zijn bij stad Herentals, maar bijvoorbeeld een IT-beheerder die de applicatie onderhoudt of een extern bedrijf dat in staat is voor het legen van vuilnisbakken. Een externe gebruiker kan dezelfde rollen worden toegewezen als een gewone gebruiker.

Een gebruiker kan ook zijn eigen profiel raadplegen. Hier kunnen bepaalde zaken worden bewerkt, zoals de voor- en achternaam, het emailadres, de username en het wachtwoord. Een gebruiker kan de rollen bekijken die hij/zij heeft toegewezen gekregen, maar kan deze niet zelf aanpassen.

Opbouw

Onze oplossing is opgebouwd uit 2 delen. Enerzijds is er de frontend applicatie, en anderzijds is er de backend API. Deze werken onafhankelijk van elkaar, maar werken hand in hand samen op een efficiënte manier.

De frontend is een Angular applicatie. Angular is een frontend framework gebaseerd op TypeScript. De inhoud van een Angular applicatie is niet statisch, maar wordt dynamisch gerenderd zonder de pagina te moeten herladen. Dit zorgt ervoor dat het een zeer snelle en performante applicatie is. Een nadeel van een Angular applicatie is dat het hierdoor zeer moeilijk is om de app te optimaliseren voor zoekmachines. Dit is echter geen groot probleem omdat de applicatie enkel gebruikt zal worden door werknemers van stad Herentals, en het dus geen publieke website is die zoveel mogelijk bezoekers zal moeten aantrekken. Via de ingebouwde service-componenten wordt de data van de backend omgezet naar overzichtelijke, leesbare informatie in de app. Daarnaast is er ook PWA-functionaliteit ingebouwd. Dit wil zeggen dat de applicatie ook als mobiele app kan geïnstalleerd worden op smartphones en tablets. Omdat het voor deze usecase nodig is om constant de laatste nieuwe data op te halen, is het jammer genoeg niet mogelijk om deze mobiele app offline te gebruiken. Een gebruiker zal dus altijd verbonden moeten zijn met een (draadloos) netwerk.

De backend is een Laravel Rest API, die in verbinding staat met een MySQL database. Laravel is een PHP framework dat gebruikt kan worden voor zowel backend, en full-stack applicaties. In onze API wordt enkel de backend functionaliteit van Laravel gebruikt. Vanuit de Angular app zullen dus requests worden gedaan naar deze backend API. Op basis van deze request zal de API een gepaste JSON response terugsturen met data uit de database waaraan hij verbonden is. Hierdoor is de frontend app vaak veel trager dan de backend API. Nadeel hiervan is dat de betekenisvolle informatie vaak langer laadt in de frontend.

De Laravel API wordt samen met de database gehost op Azure:

<https://laravel-smartcity.azurewebsites.net/>
<https://github.com/UpSmartOrg/UpSmartAppApi>

De frontend applicatie wordt gehost op een bestaand hostingpakket van een teamlid:

<http://smartcity.seppealaerts.be>
<https://github.com/UpSmartOrg/UpSmartApp>

Installatie

De applicatie is volledig online beschikbaar. Als u de projecten lokaal wil laten draaien, zijn er eerst een paar installaties nodig.

- Algemeen
 1. Installeer GIT
<https://git-scm.com/downloads>
 2. Installeer nodeJS (versie 10.13.0 of later)
<https://nodejs.org/en/download/>
- Backend
 1. Installeer PHP (versie 7.4 of later)
<https://www.php.net/downloads.php>
 - Pak het zip-bestand uit in de root van je C-drive en hernoem de map naar: C:\PHP74
 - Open "Omgevingsvariabelen van het systeem bewerken"

- Klik op “Omgevingsvariabelen”
- Kies onder “Systeemvariabelen” de variabele “Path” en klik op “Bewerken”
- Klik op “Nieuw” en voeg toe: C:\PHP74
- Dupliceer de file C:\PHP74\php.ini-development en rename de file naar php.ini
- Open php.ini en haal de ‘;’ voor volgende extensies weg:

```
extension_dir = "ext"
```

```
extension=curl
```

```
extension=fileinfo
```

```
extension=gd2
```

```
extension=mbstring
```

```
extension=odbc
```

```
extension=openssl
```

```
extension=pdo_mysql
```

```
extension=sockets
```

- Vervang de 128M memory limit met -1 (unlimited):

```
memory_limit = -1
```

2. Installeer Composer
<https://getcomposer.org/Composer-Setup.exe>
 3. Clone de GitHub repo
<https://github.com/UpsmartOrg/UpsmartAppApi>
 4. Open een PowerShell venster in de geclonede directory
 5. `npm install`
 6. `npm start`
 7. Na het bouwen zal de app geopend worden (bv. op localhost:8000)
- Frontend
 1. Installeer Angular CLI
 - Open een PowerShell venster als administrator
 - `npm install -g @angular/cli@10.0.2`
 2. Clone de GitHub repo
<https://github.com/UpsmartOrg/UpsmartApp>
 3. Open een PowerShell venster in de geclonede directory
 4. `npm install`
 5. Vervang de urls in de services naar de url van de lokale backend (bv. localhost:8000)
 6. `npm start`
 7. De app wordt na het bouwen geopend op localhost:4200

Admin Dashboard



Zoeken					Kies rol Alle Rollen ▾			
#	Voornaam	Familienaam	Emailadres	Username	Rollen	Externe	Aangemaakt op	Acties
1	Aron	Arkens	AronA@herentals.be	aron.a	Gro	Nee	01-01-2020	
2	Barry	Bostols	BarryB@herentals.be	barry.b	Par	Nee	01-01-2020	
3	Connie	Cannaerts	ConnieC@herentals.be	connie.c	Com	Nee	01-01-2020	
4	Dirk	Dockers	DirkD@herentals.be	dirk.d	Gro, Par, Com, Adm	Ja	01-01-2020	
5	Erica	Erentals	EricaE@herentals.be	erica.e	Adm	Nee	01-01-2020	
GEBRUIKER TOEVOEGEN								

Profiel



Details

Voornaam	Familienaam
Dirk	Dockers
Emailadres	Username
DirkD@herentals.be	dirk.d
Aangemaakt op	
01-01-2020	
<input type="checkbox"/> Admin <input type="checkbox"/> Groendienst <input type="checkbox"/> Participatie <input type="checkbox"/> Communicatie	
<input checked="" type="checkbox"/> Extern	
WIJZIG PROFIEL	

Participatie Dashboard



Handleiding Participatie Dashboard

Op deze pagina vindt u een overzicht van alle bestaande enquêtes.

Met onderstaande zoekbalk kan u een specifieke enquête zoeken op basis van de naam, of kan u de enquêtes filteren met de dropdownlist op basis van de gebruiker die de enquête heeft aangemaakt.

De start- en einddata's tonen aan in welke periode een enquête op de kiosk beschikbaar zal zijn om in te vullen door de burgers. Bij een snelle bevraging kunnen de vragen op de homepagina van de kiosk worden getoond, en snel worden ingevuld door de gebruiker.

Via de acties aan de rechterzijde van de tabel kan u een enquête aanpassen of volledig verwijderen.

Ten slotte kan u een nieuwe enquête aanmaken via de knop "enquête toevoegen" onderaan.

Zoeken

Aangemaakt door
Alle gebruikers

#	Titel	Startdatum	Einddatum	Snelle bevraging	Aangemaakt door	Acties
1	Enquête over parkeren in Herentals.	31-12-2020	30-07-2021	Ja	Barry Bostols	
2	Covid enquête	01-09-2020	31-03-2021	Nee	Connie Cannaerts	
3	Vragen omtrent activiteiten	01-05-2021	31-08-2021	Nee	Barry Bostols	

ENQUETE TOEVOEGEN



Participatie Kiosk

Welkom op de participatiekiosk van stad Herentals, mede mogelijk gemaakt door Upsmart. In het stadsnieuws kan u allerhande info vinden over Herentals. Daarnaast kan u ook een (korte) bevraging invullen, waardoor stad Herentals zijn diensten kan verbeteren en kan afstemmen op de noden van de inwoners.

SNELLE BEVRAGING

1. Hebt u vaak problemen om parkeerplaats te vinden?

- ☐ Ja
☐ Neen

OPSLAAN EN VERZENDEN

ALLE BEVRAGINGEN

HERENTALS STADSNIEUWS

Rioleringswerken Lierseweg

17-02-2021

Vanaf maandag 15 tot woensdag 17 februari vinden er rioleringswerken plaats de Lierseweg. De werkzaa...

[Verder lezen](#)

ALLE BERICHTEN

Besluit

Tijdens het ontwikkelen van dit project hebben wij als team veel bijgeleerd. Dit zowel op hard als soft skills. De technologieën met elkaar combineren en gezamenlijk integreren in ons systeem was enkel mogelijk dankzij een goede samenwerking binnen het team. Daar we dit project volledig digitaal realiseerden, gebruikten we voornamelijk Microsoft Teams om met elkaar te communiceren. De taken werden verdeeld volgens de agile principes van sprints, epics en user stories. Om deze planning digitaal te kunnen opvolgen gebruikten we de tool Monday.com. Tijdens de wekelijkse sprint meetings met de opdrachtgever gebruikten we deze momenten om actief feedback te krijgen over het project en deze tijdens de volgende sprints te implementeren in het systeem.

We zijn zelf zeer trots op het resultaat. Door dit proef-of-concept te realiseren hopen we onze opdrachtgever te kunnen overtuigen van een uitrol in hun gemeente. De investering van de LoRa infrastructuur kan een springplank zijn naar veel meer slimme objecten in de stad. Het vuilnisbakken systeem maakt het mogelijk om gemeentediensten efficiënter te laten werken. Zo kunnen zij een lijst via de applicatie verkrijgen van enkel de volle vuilnisbakken die ze de volgende dag moeten ophalen.

Het kiosk systeem maakt het anderzijds mogelijk om ook de burger gebruik te laten maken van de digitale diensten van de stad. Door de enquêtes op dit platform kan de stad achterhalen via de dashboards in de applicatie wat de meningen zijn van de burger. Ook van deze kiosk maakte we een slim toestel dat data verstuurd en geïntegreerd werd in onze applicatie.