Build your first social media Dashboard in minutes with Node-RED and Db2 for i or PostgreSQL

Benoit Marolleau Solution Architect
benoit.marolleau@fr.ibm.com
Twitter @MarolleauBenoit
Linkedin https://www.linkedin.com/in/bmarolleau/

Before Starting

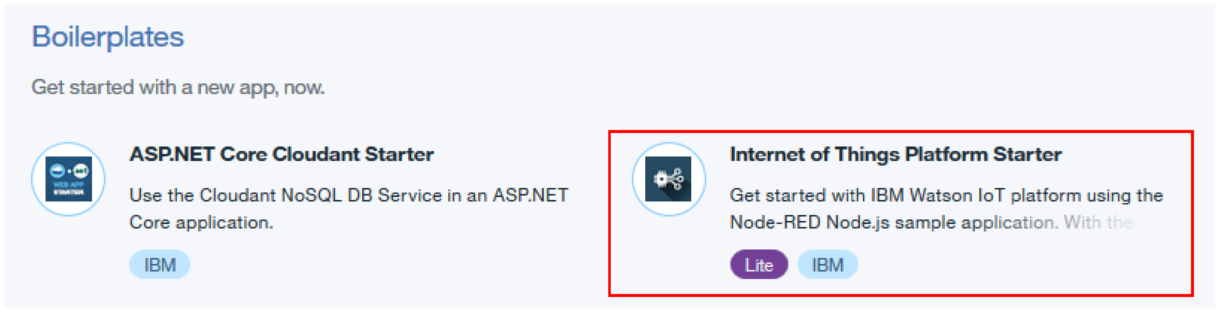
This Lab was designed to be used on premise with the IBM i operating system and Db2 for i as illustrated in the online version. Online version in DeveloperWorks:

https://www.ibm.com/developerworks/ibmi/library/i-social-media-dashboard/

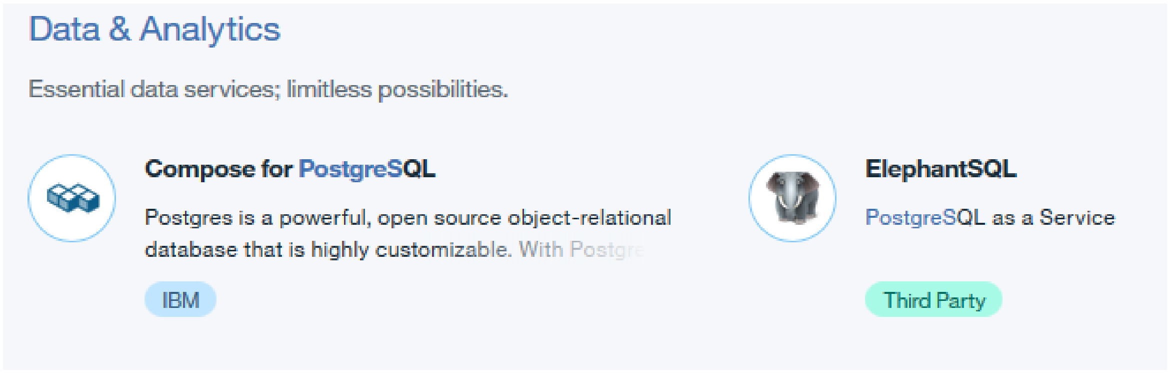
If you don't have the chance & privilege to have an IBM i (best operating system ever) or if you run the lab online (IBM Cloud), you can replace the "Db2 for i" node by a standard SQL compatible database node, like ElephantDB, PostgreSQL by Compose, etc.

The Postgres SQL syntax is really close to Db2 SQL. The SQL queries is provided in the lab content.

1) First, You will have to create a Node-RED application using the IoT Starter boilerplate available in the IBM Cloud catalog (Free Plan)


Boilerplates
Get started with a new app, now.
ASP.NET Core Cloudant Starter
Use the Cloudant NoSQL DB Service in an ASP.NET Core application.
IBM
Internet of Things Platform Starter
Get started with IBM Watson IoT platform using the Node-RED Node.js sample application. With the
Lite
IBM

2) Create a database service. You can use the following services from the IBM Cloud catalog for example, and use the administration console of the Database service of your choice to create the table and triggers, insert data in the database, etc. Prefer a PostgreSQL based service for SQL Compatibility reasons.


Data & Analytics
Essential data services; limitless possibilities.
Compose for PostgreSQL
Postgres is a powerful, open source object-relational database that is highly customizable. With Postgre
IBM
ElephantSQL
PostgreSQL as a Service
Third Party

3) Bind the database service to your Node-Red application from the IBM Cloud console.

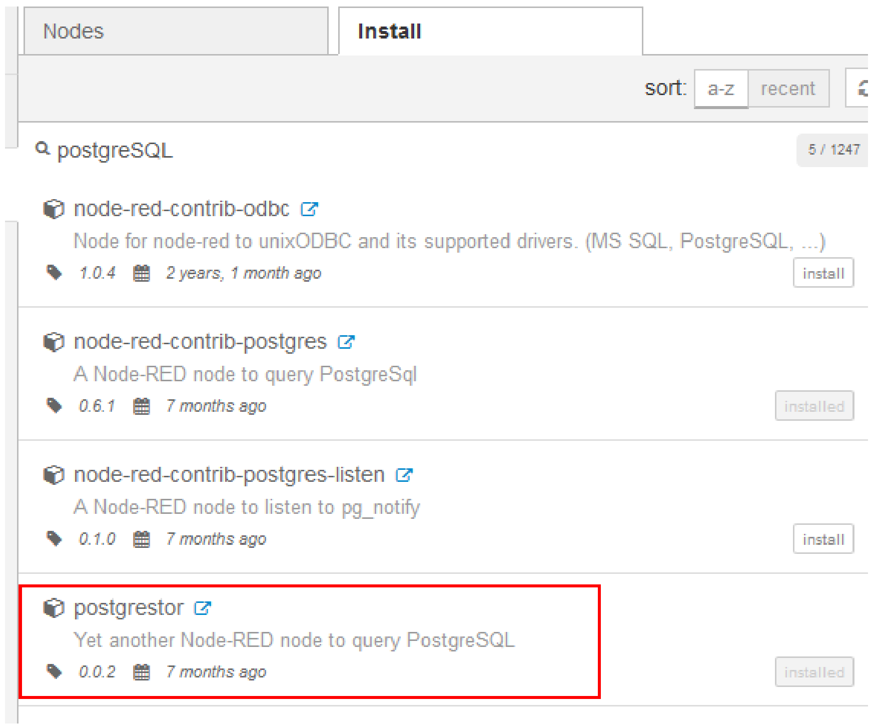
4) Optional: Install a DB client of your choice on your laptop for creating the Tweets table & trigger, and insert the test data.
For Db2 for i, download and use the free Access Client Solution (ACS).

   For PostgreSQL, use https://www.pgadmin.org/download/

   If you have a cloud based PostgreSQL service, you can also use the out of the box SQL interface.

5) Write down the hostname & credentials of your database service from the PostgreSQL Service running in the IBM Cloud and create a connection from your favorite db client. These credentials will be used in your node-red PostgreSQL nodes as well.

6) Install a PostgreSQL node in the Node-Red palette from the top right menu button. This node will be used instead of the "Db2 for I" node for querying your database.



node-red-contrib-odbc
Node for node-red to unixODBC and its supported drivers. (MS SQL, PostgreSQL, ...)
1.0.4 2 years, 1 month ago install

node-red-contrib-postgres
A Node-RED node to query PostgreSql
0.6.1 7 months ago installed

node-red-contrib-postgres-listen
A Node-RED node to listen to pg_notify
0.1.0 7 months ago install

postgrestor
Yet another Node-RED node to query PostgreSQL
0.0.2 7 months ago installed

Most the PostgreSQL and Db2 instructions are the same, please adapt the content based on your database engine. The Db2 for i node has to be replaced by the PostgreSQL node.

Introduction

How to easily and rapidly create business values from open source technology, cloud services, and your data on IBM i? Make your idea a reality by prototyping like a master using open source technologies, Node-RED, and Node.js in conjunction with IBM® Db2® for i!

In this article, we implement (with almost no lines of code and in a few minutes) a simple social media dashboard, including sentiment analysis, natural language processing (NLP) and business data enrichment. Stop talking and simply do it!

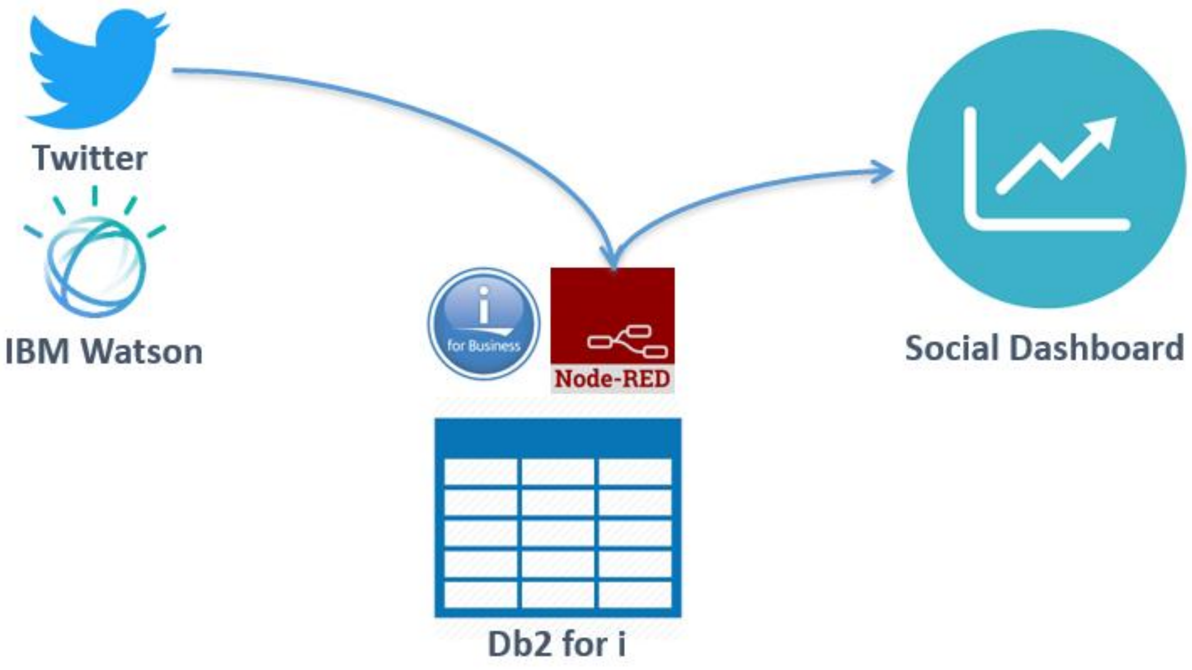


Figure 1. Social media dashboards using Node-RED, IBM Watson, and Db2 for i

Social listening is the process of monitoring the conversations and mentions of your brands online, and is valuable for all organizations, marketing teams, customer support teams, business development, and competition surveys, especially when used and mixed with business data.

This process allows you to get real-time insights in many use cases. Some of them are listed below.

- How your brand is mentioned in the social media
- How you can adjust marketing campaigns based on metrics such as impressions and engagements

- How you can have more information about your client segmentation
- How to react to customer complaints in real time

The objective here is to highlight the capabilities of open source technologies available on IBM i (integrated with Db2 for i), and illustrate how to get started easily with digital transformation, cognitive computing, APIs, and hybrid cloud.

Node-RED is an open source flow-based coding tool coming from Internet of Things (IoT), brought by IBM Emerging Technology, proposing a large number of nodes from the node-red.org community. Besides IoT, it is ideal for building micro-services based architecture solutions, prototyping, or creating and consuming cloud or on-premises APIswith no lines of code.

Need more information about Node-RED on IBM i and how to get started? Refer IBM developerWorks article, Running Node-RED on IBM i: Installation and first flow.

This article explains first how to implement real-time sentiment analysis for particular hashtags and write this information in your Db2 database, involving a **Twitter** node, optionally a **Watson Language Translation** node, a sentiment analysis node, and a **Db2 for i** node to enrich business data.

It also describes how to visualize data stored in Db2 for i, creating a simple social media dashboard Web application, using dashboard' nodes and 'Db2 for i' nodes.

The optional Watson™ *Language Translation* node will identify languages and translate tweets in English on the fly before sentiment analysis.

What you'll need

To write this social media dashboard, you need the following minimum prerequisites:

IBM **Db2**

- 5733OPS option 10 that is, Node.js v6, Node-RED installed, and a web browser. Refer to the “Running Node-RED on IBM i: Installation and first flow” article to view all the detailed prerequisites.
- A decent Db2 for i SQL editor connected to an IBM i 7.2 system and below. You can use IBM i Access Client Solutions and the embedded SQL editor.

PostgreSQL

- Node.RED up & running: either from IBM Cloud, on a linux/windows server, or on your laptop.
- A PostgreSQL database server running (Cloud or on your laptop if you run this lab locally, Node.RED and DB. A decent SQL editor connected to a PostgreSQL.

IBM **Db2** PostgreSQL

- A Twitter account, and a connection to the Internet from IBM i. Optionally, you can simulate tweets using the test data provided in this article.
- Optional: An IBM Cloud (former IBM Bluemix®) account for using IBM Watson and a Watson *Language Translation* node.
  You can register for a free trial or freemium account or log in to IBM Cloud if you already have an account.

Start Node-RED

Refer to this article if you want to know more about Node-RED and how to get started on IBM i. Start Node-RED and connect to it for example, using the default URL: http://<IBMI-IP>:1880 where <IBM-IP> is the IP address of your IBM i system.

It is a good practice to set authentication credentials to your Node-RED and switch from HTTP to HTTPS.

Create a basic tweets table and import sample data

Let us consider an example where we store only a subset of the information contained in the incoming tweets, but it is good to notice that Twitter APIs return a lot more attributes in a JSON object for each tweet, as documented on the Twitter Developer Platform website.

Also note that the Node-RED Twitter node returns all this information in JSON, and we will parse and then store this data in a standard Db2 table for the sake of simplicity. However, it is better to use Db2 for i for directly storing the incoming tweets in JSON in a CLOB column.

You can then retrieve information about tweets with the powerful JSON_TABLE functions available on Db2 for i.

1. Create a simple **Tweets** table for storing our Twitter activity. For that purpose, open a SQL editor, for example IBM i Access Client Solutions and run the following SQL statements for creating a sentiment library, a **Tweets** table, and a simple trigger for that table.

```
CREATE collection SENTIMENT;
```

```
CREATE OR REPLACE TABLE SENTIMENT.TWEETS (

    id SMALLINT NOT NULL

        GENERATED ALWAYS AS IDENTITY

        (START WITH 1

        INCREMENT BY 1

        CYCLE),

    TWEET VARCHAR(5000),

    SCREEN_NAME VARCHAR(255),

    LOCATION VARCHAR(255),

    SENTIMENT_SCORE INT,

    SENTIMENT_GROUP VARCHAR(255),

    TIMESTP TIMESTAMP,

    TIMESTAMP_MS VARCHAR(255),

    primary key (id)

);
```

```
CREATE OR REPLACE TRIGGER SENTIMENT.Tweets_sentiment

  BEFORE INSERT OR UPDATE ON SENTIMENT.Tweets

  REFERENCING NEW AS n OLD AS o

  FOR EACH ROW MODE Db2ROW

  BEGIN

    SET n.SENTIMENT_GROUP = CASE WHEN n.SENTIMENT_SCORE > 0 THEN 'positive'

                               WHEN n.SENTIMENT_SCORE = 0   THEN 'neutral'

                                 WHEN n.SENTIMENT_SCORE < 0  THEN 'negative'

    END;

 END;
```

```
CREATE schema SENTIMENT;
```

```
CREATE TABLE sentiment.TWEETS (

    id SERIAL NOT NULL PRIMARY KEY   ,

    TWEET VARCHAR(5000),

    SCREEN_NAME VARCHAR(255),

    LOCATION VARCHAR(255),

    SENTIMENT_SCORE INT,

    SENTIMENT_GROUP VARCHAR(255),

    TIMESTP TIMESTAMP,

    TIMESTAMP_MS VARCHAR(255)

);
```

```
CREATE OR REPLACE FUNCTION sentiment.updateSentiment() RETURNS TRIGGER AS
$example_table$

   BEGIN

     if NEW.SENTIMENT_SCORE > 0 THEN

             NEW.SENTIMENT_GROUP := 'positive';

    ELSEIF NEW.SENTIMENT_SCORE = 0   THEN

             NEW.SENTIMENT_GROUP :=  'neutral';

    ELSE

             NEW.SENTIMENT_GROUP := 'negative';

  END IF;

    return NEW;

   END

$example_table$ LANGUAGE plpgsql;


CREATE TRIGGER Tweets_sentiment before INSERT ON sentiment.TWEETS

FOR EACH ROW EXECUTE PROCEDURE sentiment.updateSentiment();
```

IBM **Db2**

If you want to use an existing library on your system, just copy the `CREATE TABLE` and `CREATE TRIGGER` sections above, and start journaling the **Tweets** table.

The `Tweets_sentiment` trigger will just update the inserted or updated record with a `SENTIMENT_GROUP` (positive, negative, neutral) depending on the `SENTIMENT_SCORE` value.

2. Let's populate this table with sample data (with 12 public tweets). Later in this article, our Twitter node will populate the Tweets table for us.

IBM **Db2**

```
INSERT INTO
SENTIMENT.TWEETS(TWEET,SCREEN_NAME,LOCATION,SENTIMENT_SCORE,TIMESTP,TIMESTAMP_MS)
VALUES('What an impressive panel at our ask the experts during our conference!
#ibmi #cshk2017 https://t.co/StgAsBPFXv','MBuurRasmussen','nolocation',3,CURRENT
TIMESTAMP,1508850681582);

INSERT INTO
SENTIMENT.TWEETS(TWEET,SCREEN_NAME,LOCATION,SENTIMENT_SCORE,TIMESTP,TIMESTAMP_MS)
VALUES('Thanks, @chris_lalevee for this bodacious article about Node-RED on #IBMi!
#IBMiOSS https://t.co/yursu7d2GJ','IBMJesseG','Rochester, MI',2,CURRENT TIMESTAMP,
1508850903650);

INSERT INTO SENTIMENT.TWEETS
(TWEET,SCREEN_NAME,LOCATION,SENTIMENT_SCORE,TIMESTP,TIMESTAMP_MS) VALUES ('RT
@IBMi: Thank you, for teaching us all a valuable lesson,
https://t.co/idje2J3','IBMiFan','United States of America',6,CURRENT TIMESTAMP,
1508851116842);

INSERT INTO SENTIMENT.TWEETS
(TWEET,SCREEN_NAME,LOCATION,SENTIMENT_SCORE,TIMESTP,TIMESTAMP_MS) VALUES('RT #IBMi
Opensource tools https://t.co/s8sjeu','ClubCEO','Texas, USA',0,CURRENT
TIMESTAMP,1508851142085);

INSERT INTO
SENTIMENT.TWEETS(TWEET,SCREEN_NAME,LOCATION,SENTIMENT_SCORE,TIMESTP,TIMESTAMP_MS)
VALUES('Excellent article about #ibmi & Db2 here
https://t.co/s8sjeu!!','MariaMe6452','Dublin, Ireland',3,CURRENT TIMESTAMP,
1508851159475);

INSERT INTO
SENTIMENT.TWEETS(TWEET,SCREEN_NAME,LOCATION,SENTIMENT_SCORE,TIMESTP,TIMESTAMP_MS)
VALUES('Meet the next generation of #IBMi and #IBMAIX with these #IBMFreshFaces
featured in @IBMPowerMag: https://t.co/PbpnGpypnE','IBMPowerSystems','Sugarland,
TX',3,CURRENT TIMESTAMP, 1508851165417);

INSERT INTO
SENTIMENT.TWEETS(TWEET,SCREEN_NAME,LOCATION,SENTIMENT_SCORE,TIMESTP,TIMESTAMP_MS)
VALUES('RT @ibm: Our Enterprise #IBMi Modernization White Paper helps you
understand how #Nodejs can attract new development talent! ','3E8TestBot','Lost in
Space',5,CURRENT TIMESTAMP, 1508851192995);

INSERT INTO
SENTIMENT.TWEETS(TWEET,SCREEN_NAME,LOCATION,SENTIMENT_SCORE,TIMESTP,TIMESTAMP_MS)
VALUES('RT @kadler_ibm: Good news, everyone! Come see me at @zendcon and we can
talk all things #IBMiOSS and #PHP on #IBMi. I might even ha ','phpc','Freenode
IRC',5,CURRENT TIMESTAMP, 1508851202542);

INSERT INTO
SENTIMENT.TWEETS(TWEET,SCREEN_NAME,LOCATION,SENTIMENT_SCORE,TIMESTP,TIMESTAMP_MS)
VALUES('RT @HelpSystemsMN: What are #IBMi special authorities &amp; who should have
them? Security expert @DaljitSingh explains &gt; ','J_Buck51','Kenosha,
WI',0,CURRENT TIMESTAMP,1508851319619);

INSERT INTO
SENTIMENT.TWEETS(TWEET,SCREEN_NAME,LOCATION,SENTIMENT_SCORE,TIMESTP,TIMESTAMP_MS)
VALUES('RT @chris_lalevee: Install @NodeRED on #IBMi (https://t.co/DHlifEAEuo) and
test Db2 for i Node !! https://t.co/KKu5Ji5ihW','MarolleauBenoit','Montpellier,
France',0,CURRENT TIMESTAMP,1508851419619);

INSERT INTO
SENTIMENT.TWEETS(TWEET,SCREEN_NAME,LOCATION,SENTIMENT_SCORE,TIMESTP,TIMESTAMP_MS)
VALUES('Looking forward to meeting #IBMi customers next week at the Jack Henry
Conference in Nashville. Stop by booth 208
https://t.co/gZs4MXNsO8','petem59','Connecticut, USA',0,CURRENT
TIMESTAMP,1508851544257);

INSERT INTO
SENTIMENT.TWEETS(TWEET,SCREEN_NAME,LOCATION,SENTIMENT_SCORE,TIMESTP,TIMESTAMP_MS)
VALUES('RT #IBMi #db2 #XML Des soucis de gestion dates/heures avec XMLTABLE ? Allez
plus loin avec les fonctions XQuery/XPath ','PhBourgeoisIBM','Paris,
France',0,CURRENT TIMESTAMP,1508851704005);
```

PostgreSQL Use the same SQL INSERT queries, replacing `CURRENT TIMESTAMP` by `CURRENT_TIMESTAMP (with underscore)`

IBM **Db2**

Install the *Db2 for i* node

First, we need to install the Db2 for i node in our Node-RED palette. You can do it by using the command line as explained on the Db2 for i node home page or by using the Node-RED graphical interface as illustrated below.

In Node-RED, click the button at the upper-right corner and click **Manage Palette.**

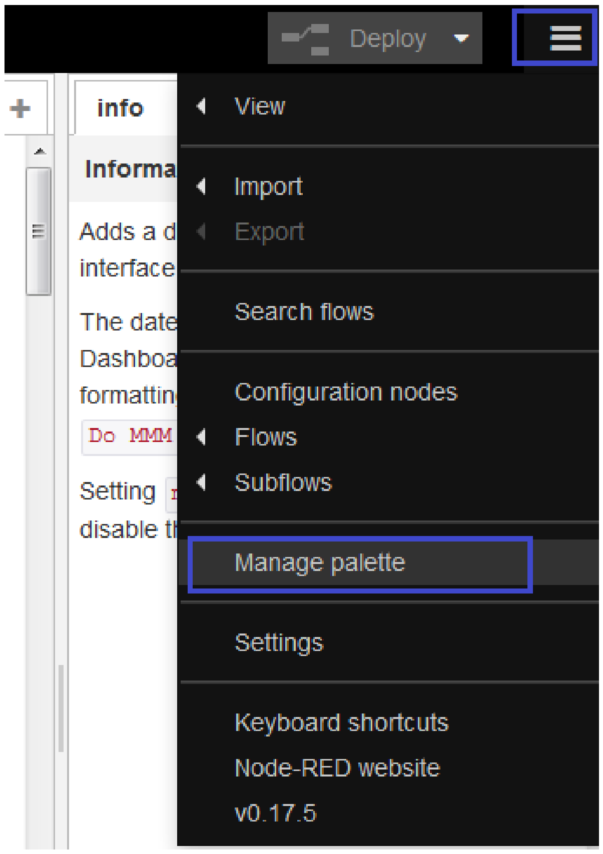
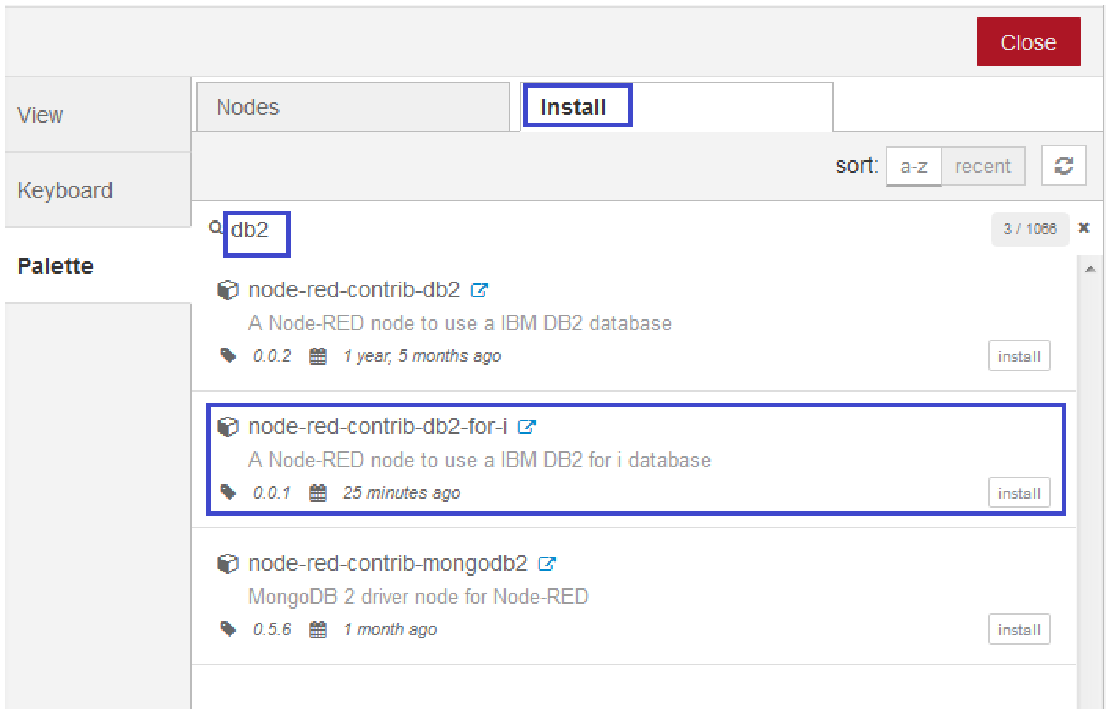


Figure 2 Add nodes to the Node-RED palette

On the **Install** tab, search for **db2**, and then search for **node-red-contrib-db2-for-i**, and click **Install**. Wait for the installation confirmation dialog box to be displayed, and then restart Node-RED.



node-red-contrib-db2
A Node-RED node to use a IBM DB2 database
0.0.2 1 year, 5 months ago install

node-red-contrib-db2-for-i
A Node-RED node to use a IBM DB2 for i database
0.0.1 25 minutes ago install

node-red-contrib-mongodb2
MongoDB 2 driver node for Node-RED
0.5.6 1 month ago install

Figure 3. Add Db2 for i "node-red-contrib-db2-for-i' nodes to the Node-RED palette

Browse the node palette on the left side and notice that the **Db2 for i** node is available in the **Storage** category.

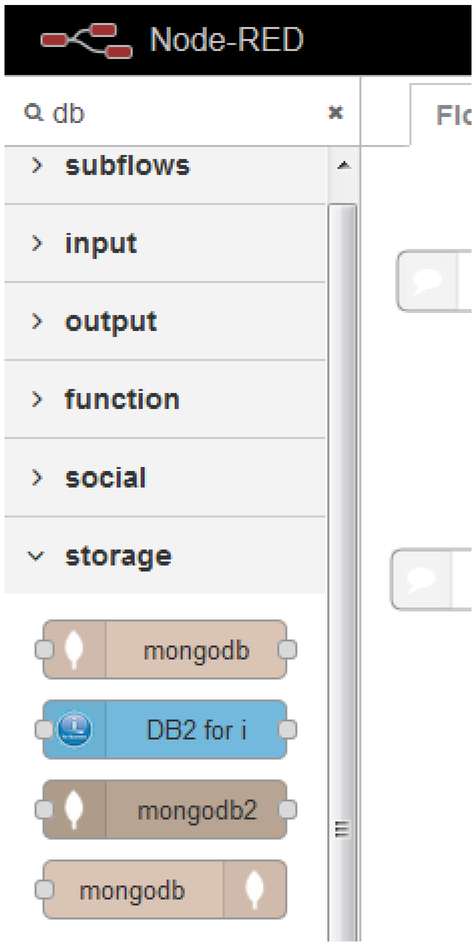
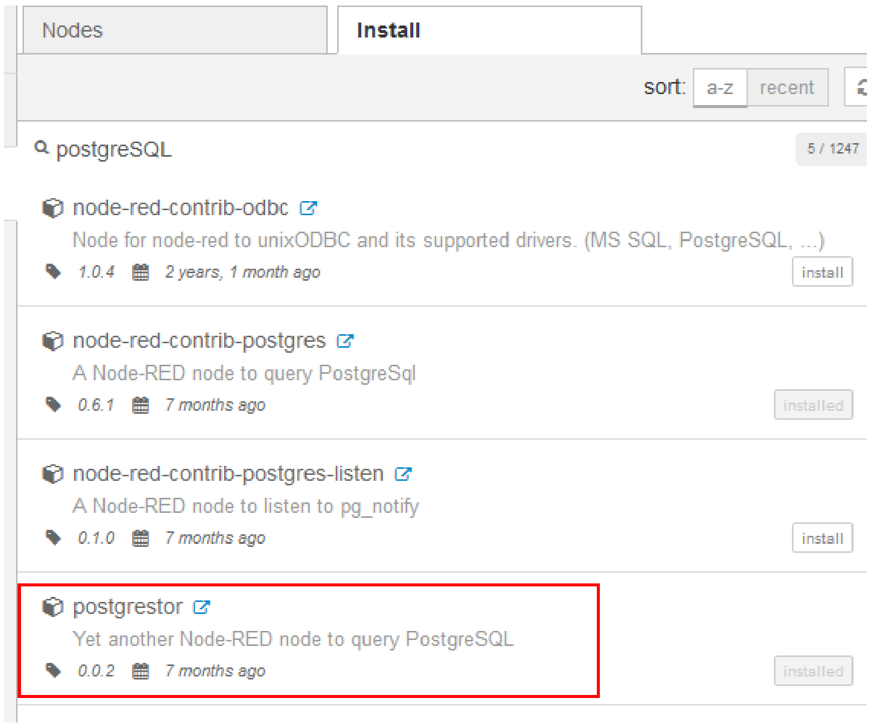


Figure 4. Db2 for i node on Node-RED

Install the *postgreSQL* node

If not already done, install the postgrestor node as shown below, and follow the same procedure than for Db2 for i above.



node-red-contrib-odbc
Node for node-red to unixODBC and its supported drivers. (MS SQL, PostgreSQL, ...)
1.0.4 2 years, 1 month ago install

node-red-contrib-postgres
A Node-RED node to query PostgreSql
0.6.1 7 months ago installed

node-red-contrib-postgres-listen
A Node-RED node to listen to pg_notify
0.1.0 7 months ago install

postgrestor
Yet another Node-RED node to query PostgreSQL
0.0.2 7 months ago installed

IBM **Db2**

Test your first flow with the *Db2 for i* node

Let’s test the node with a simple query. Drag the **Db2 for i** node to a new flow, as shown in the following figure. Then, add **SQL Query** as the inject node and **msg.payload** as the debug node.

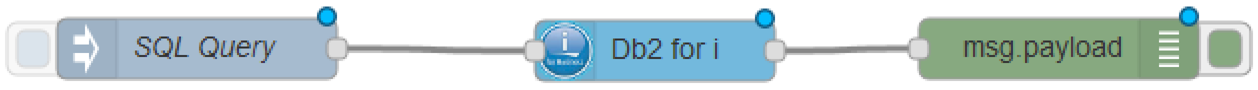
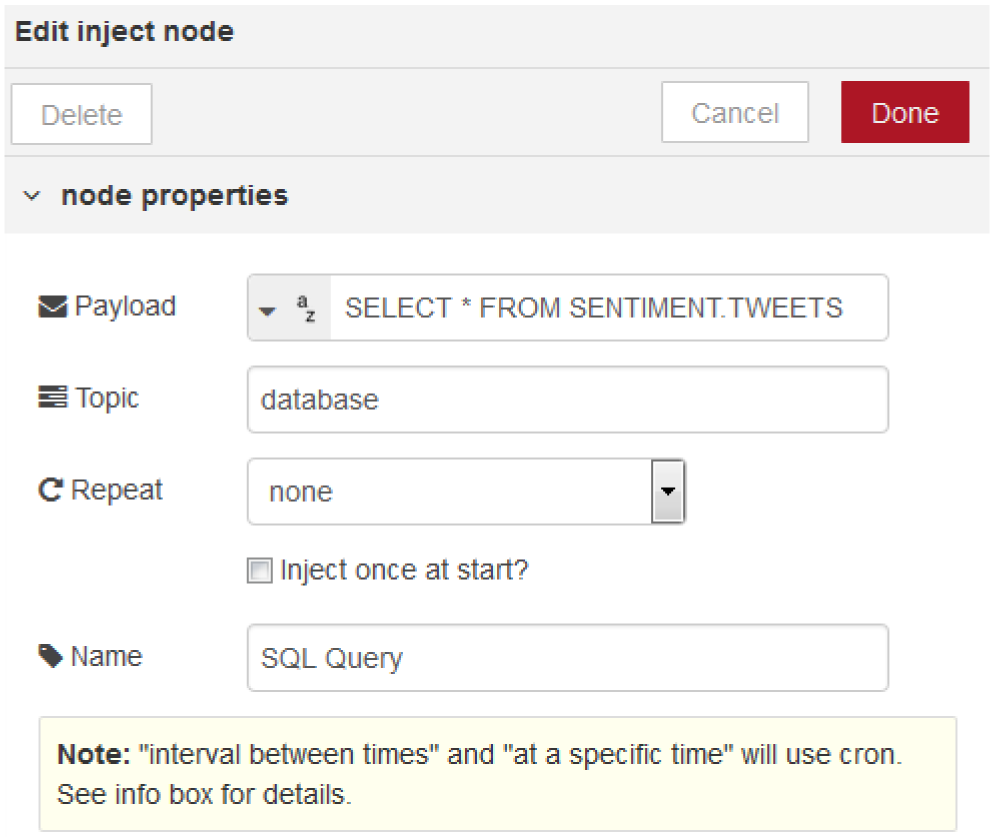


Figure 5. Simple Node-RED flow: Input Inject node – Db2 for i node – Output Debug node

1. Configure the **Inject** node with the settings and SQL query as shown in the following figure:


Edit inject node

Payload: SELECT * FROM SENTIMENT.TWEETS

Topic: database

Repeat: none

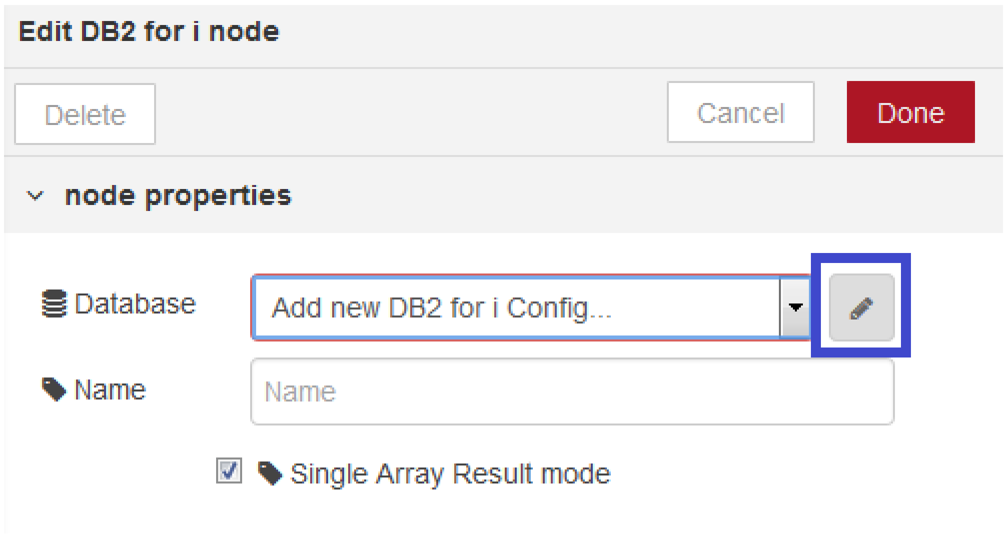
☐ Inject once at start?

Name: SQL Query

Note: "interval between times" and "at a specific time" will use cron.
See info box for details.

Figure 6. Inject node setup with a SQL query

2. Configure the **Db2 for i** node with the settings as shown in the following figure:


Edit DB2 for i node

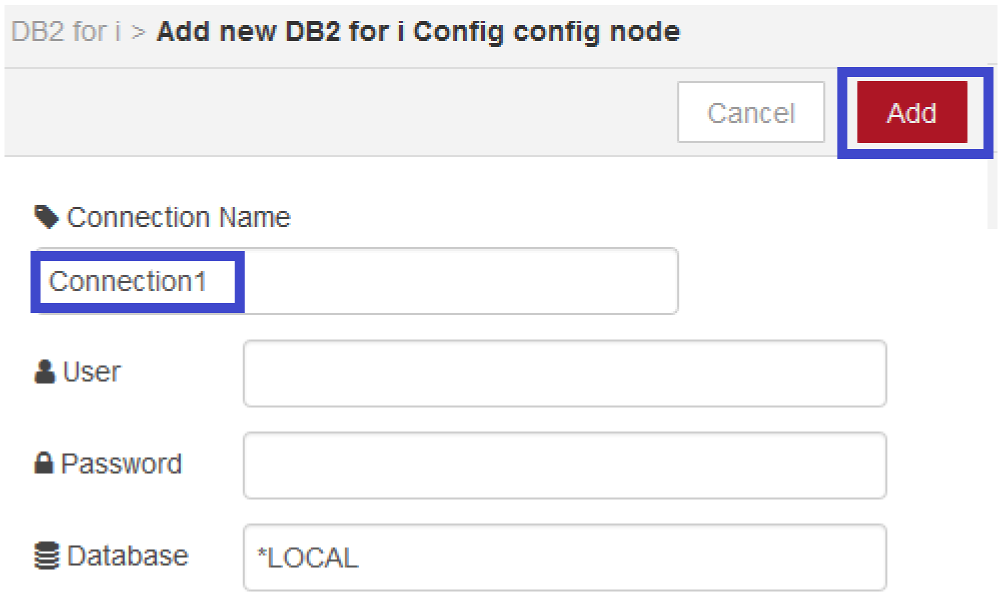
Database: Add new DB2 for i Config...

Name: Name

☑ Single Array Result mode

Figure 7. Db2 for i node settings - adding a Db2 connection

3. Set up a configuration node by specifying a connection name and, optionally, a user name/password if you don't want to use the current user profile running Node-RED, and you want to specify a particular user profile for connecting your database. Then click **Add** and **Done**.



DB2 for i > **Add new DB2 for i Config config node**

Connection Name Connection1

User

Password

Database *LOCAL

Figure 8. Configure a "Db2 for i Config" node with optional credentials

4. Test this simple flow by clicking **Deploy** and then click the **Inject node** button on the left side of your flow as shown in the following figure.

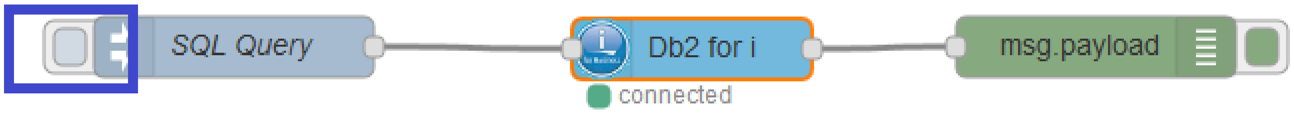
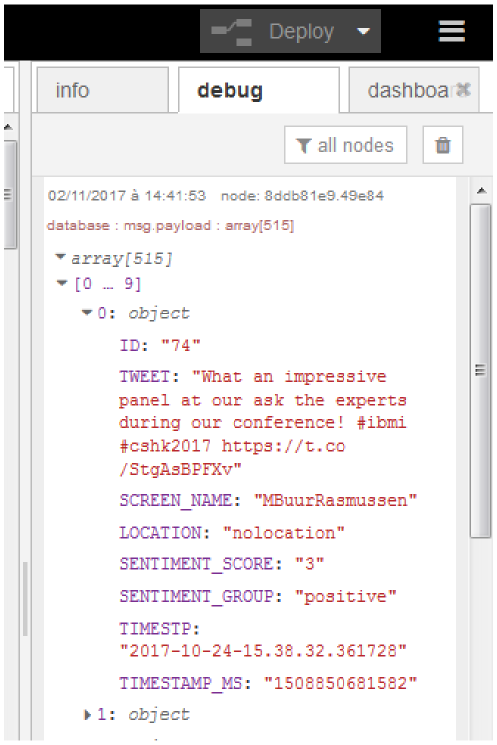


Figure 9: First Node-RED flow injecting a SQL query to a Db2 for i database

Look into the results (JSON array of rows) of the SQL query in the **Debug** panel.



```
02/11/2017 à 14:41:53   node: 8ddb81e9.49e84
database : msg.payload : array[515]
▾array[515]
 ▾[0 … 9]
   ▾0: object
      ID: "74"
      TWEET: "What an impressive
      panel at our ask the experts
      during our conference! #ibmi
      #cshk2017 https://t.co
      /StgAsBPFXv"
      SCREEN_NAME: "MBuurRasmussen"
      LOCATION: "nolocation"
      SENTIMENT_SCORE: "3"
      SENTIMENT_GROUP: "positive"
      TIMESTP:
      "2017-10-24-15.38.32.361728"
      TIMESTAMP_MS: "1508850681582"
   ▸1: object
```

Figure 10. Db2 for i JSON result set in the Node-RED debug panel

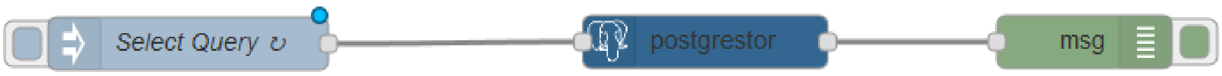
At this step, if you see no result and if you are sure that your table exists and is not empty, check if the user profile used has read access (with appropriate SQL grant or IBM i authority).

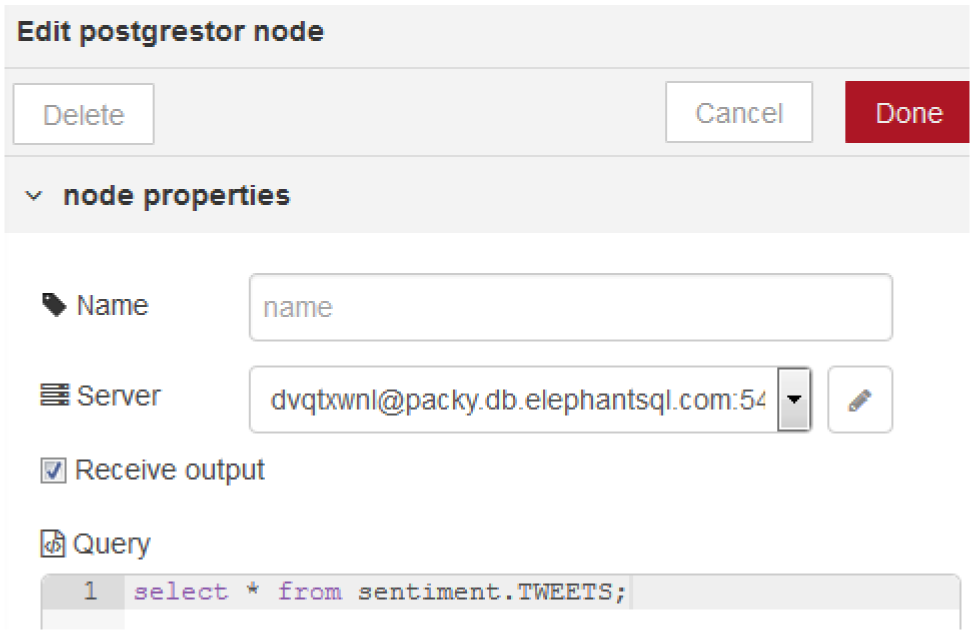
You've just written your first Node-RED flow using Db2 for i !

PostgreSQL

Test your first flow with the *PostgreSQL* node

Build the following flow with an injection node, postgrestor and debug nodes. Deploy and click the left button on the injection node to run the query.





Name: name
Server: dvqtxwnl@packy.db.elephantsql.com:54
Receive output: ✓
Query

```
select * from sentiment.TWEETS;
```

Call Twitter APIs and feed your database with tweets

The objective here is to make your IBM i capable of listening to tweets and analyzing sentiment on the fly before inserting them into your Db2 for i database.

To do so, you just have to manually drag and configure four nodes in your Node-RED flow editor.

To do this quickly, import an existing flow into your current Node-RED flow:

1. Click the menu at the upper-right corner and click **Import → Clipboard**.

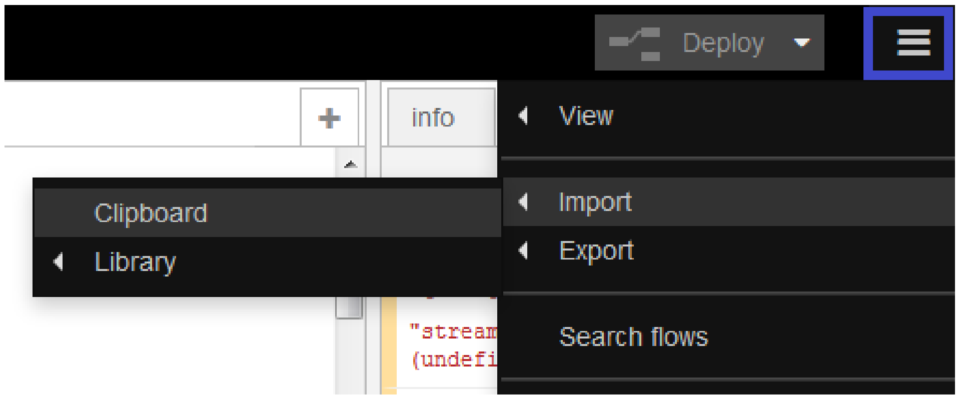


Figure 11. Flow import from clipboard in Node-RED

2. Paste the code, and click **current flow**. Then click **Import**.

IBM **Db2** https://www.ibm.com/developerworks/ibmi/library/i-social-media-dashboard/1_ibmi_twitter_listening.zip

PostgreSQL https://ibmcloud-watson-day.mybluemix.net/files/1_postgresSQL_twitter_listening.json

You should get the following result.

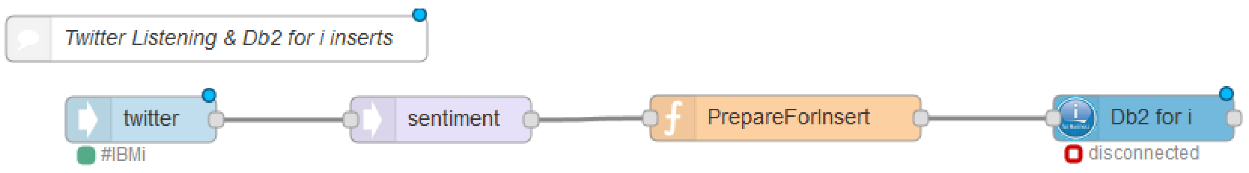


Figure 12: First Twitter listening Node-RED flow with sentiment analysis and insertion in Db2 for i

3. Configure your nodes.
    - **Twitter node**: Specify your Twitter user account and hashtag to perform a search. If you don't have a Twitter account, create one or skip this step.
    - ***Db2 for i*** **node**: Specify connection information as explained in the previous step.
    - **Function** node (`PrepareForInsert`): Control the schema and table name reflect in your environment in the JavaScript code by double-clicking on that node and modify the JavaScript function if necessary.

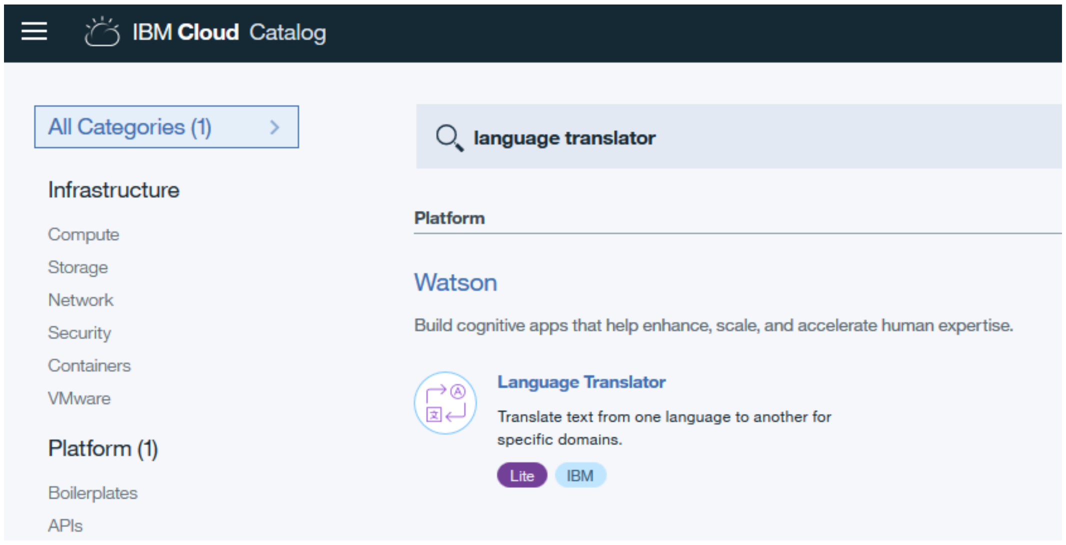
4. Click **Deploy** and wait of Twitter activity. You can optionally add a **Debug** node in your flow to see the Twitter activity in the Debug view.

5. Verify that your Tweets table is being populated by querying your table occasionally, either from your favorite SQL editor or from a Node-RED flow.

Optional: On the fly tweets translation using IBM Watson services

We can enrich incoming tweets with IBM Watson services, for example, to translate the incoming tweets into English on the fly before performing sentiment analysis. This part is optional, so you can possibly go to the dashboard creation section directly.

1. Instantiate a Watson Language Translator service from the IBM Cloud catalog, and also specify the generated credentials (user name, password, and endpoint). You'll need them when configuring the associated node in Node-RED.

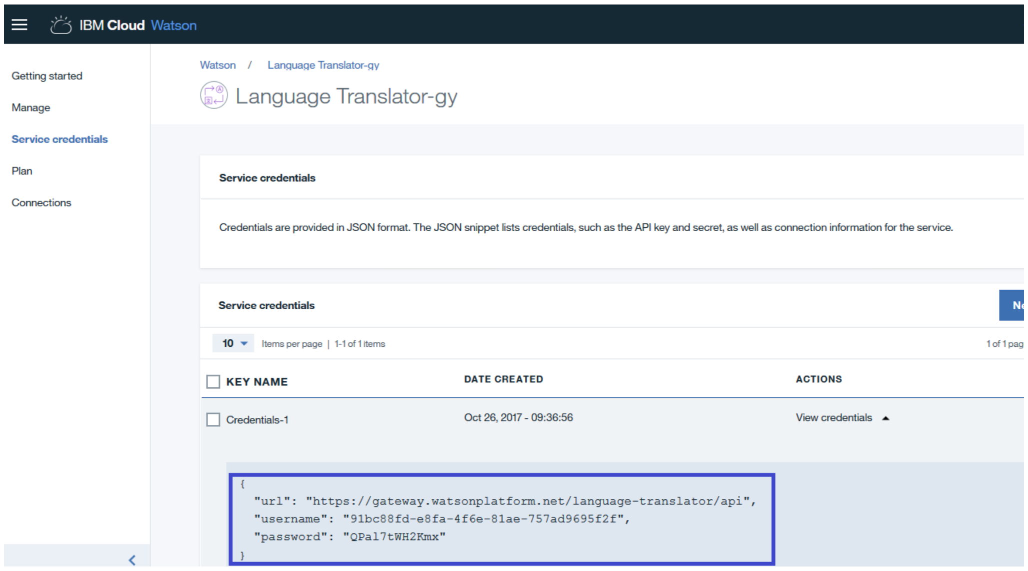


Build cognitive apps that help enhance, scale, and accelerate human expertise.

Language Translator

Translate text from one language to another for specific domains.

Figure 13. IBM Watson Language Translator service in the IBM Cloud catalog



Service credentials

Credentials are provided in JSON format. The JSON snippet lists credentials, such as the API key and secret, as well as connection information for the service.

Service credentials

10 Items per page | 1-1 of 1 items 1 of 1 pag

KEY NAME	DATE CREATED	ACTIONS
Credentials-1	Oct 26, 2017 - 09:36:56	View credentials

```
{
  "url": "https://gateway.watsonplatform.net/language-translator/api",
  "username": "91bc88fd-e8fa-4f6e-81ae-757ad9695f2f",
  "password": "QPal7tWH2Kmx"
}
```

Figure 14. IBM Watson Language Translator service credentials

2. Copy the following sample flow to your existing Twitter flow and configure your nodes with your credentials.
   a. Copy the following code in your clipboard:

https://www.ibm.com/developerworks/ibmi/library/i-social-media-dashboard/2_ibmi_watson_translator_subflow.zip

   b. In your Node-RED editor in the menu at the upper-right corner, click **Import** → **Clipboard**. Paste the code, and click **New Flow** and then click **Import**.
   This will import a Node-RED subflow. Now, add an additional node in your palette on the left side of your editor as shown below.
   A Node-RED subflow is appropriate when you want to make a portion of a flow reusable for other flows, without having to duplicate it.
   Click **Deploy** for saving and deploying the imported subflow to your Node-RED workspace.

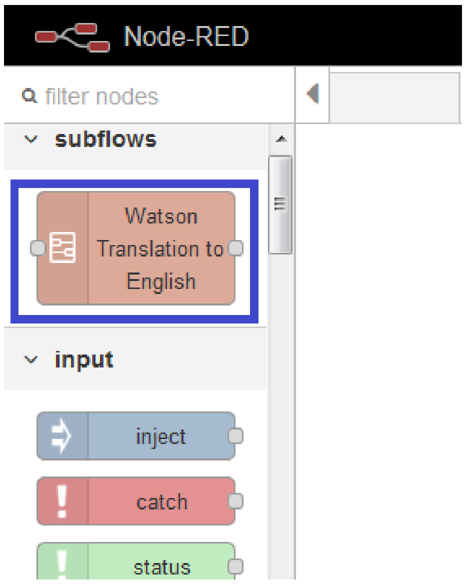


Figure 15. Creation of an IBM Watson-based subflow in Node-RED

c. Double-click **Watson Translation to English** to edit the newly created subflow, including the IBM Watson nodes.

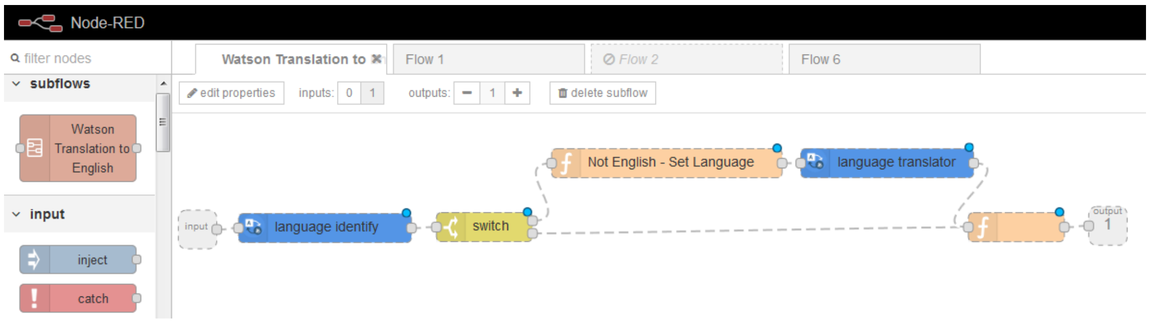
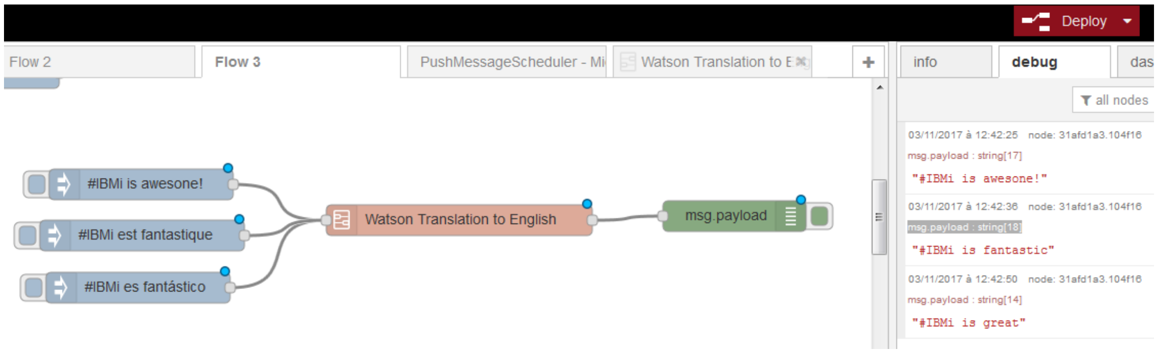


Figure 16. Details of an IBM Watson-based subflow for language identification and translation

This subflow first identities the languages used in the input message, and translates it if different from English, using IBM Watson services and APIs.

d. Configure the two Watson nodes (by specifying the user name, password, and endpoint if needed), each pointing to the same IBM Watson service you have just instantiated on IBM Cloud, but each using a different API function (language identity and language translation).

e. Test the subflow with a test flow, for example, using the manual injection nodes, as illustrated in the following figure.



```
03/11/2017 à 12:42:25   node: 31afd1a3.104f16
msg.payload : string[17]
"#IBMi is awesone!"
03/11/2017 à 12:42:36   node: 31afd1a3.104f16
msg.payload : string[18]
"#IBMi is fantastic"
03/11/2017 à 12:42:50   node: 31afd1a3.104f16
msg.payload : string[14]
"#IBMi is great"
```

Figure 17. Test of your first IBM Watson Language Translator Node-RED flow on IBM i

Thanks to IBM Watson, Your IBM i is now translating French, Spanish, and many other languages, into English!

3. Integrate this flow within your existing Twitter flow by connecting it graphically, as show in the following figure.

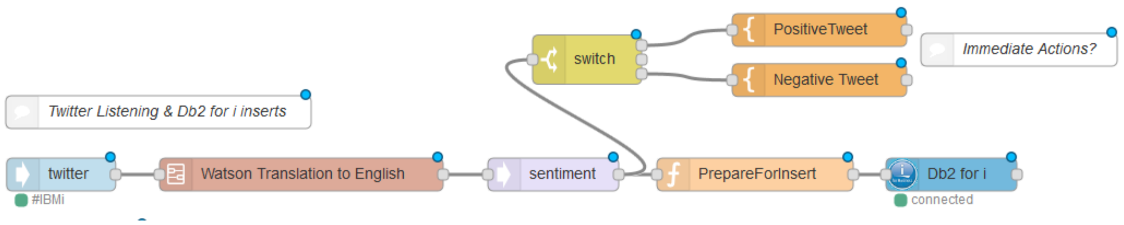


Figure 18. End-to-end Node-RED flow with social media listening, IBM Watson translation, and Db2 for i

4. Test the result either by injecting an input test payload (using the inject node) or by waiting for a new incoming tweet.

   Your IBM i is now doing Social Media listening and translating social media data on the fly with IBM Watson before inserting them into your Db2 for i database!

Visualize your Db2 for i / PostgreSQL data and social media data with a simple dashboard

In this section, we are, without coding, going to create a social media dashboard, mixing your business data on Db2 for i with data coming from the IBM Cloud and IoT.

1. Install the dashboard nodes.

   Many Node-RED exist for generating UI and graphs. The **node-red-dashboard** is one of my favorite.
   From your Node-RED editor, in the upper-right menu, click **Manage palette**, and search for **node-red-dashboard**. Then, click **Install** and restart Node-RED.

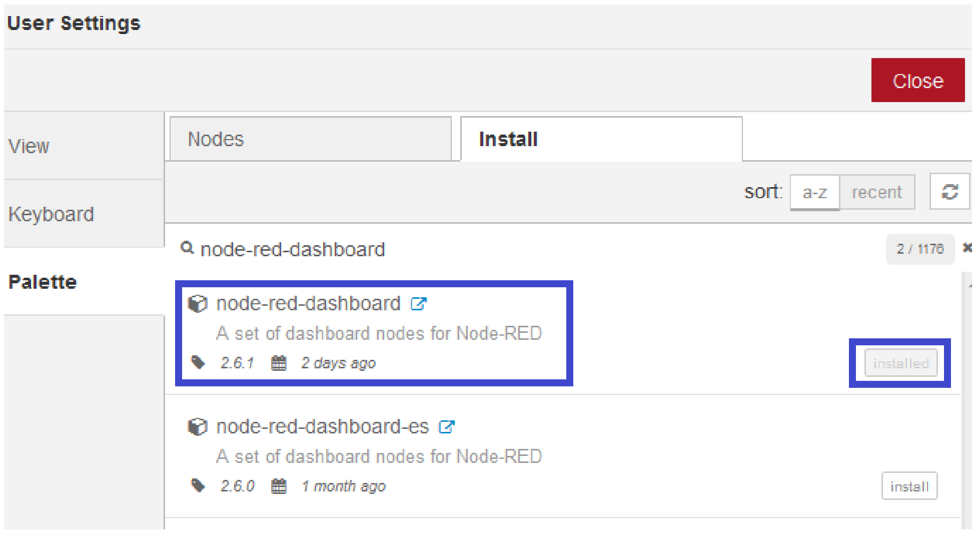


Figure 19. Installation of "node-red-dashboard" in Node-RED

2. Import the following code in the previously created Node-RED flow:

IBM Db2 https://www.ibm.com/developerworks/ibmi/library/i-social-media-dashboard/3_ibmi_simple_dashboard.zip

PostgreSQL https://ibmcloud-watson-day.mybluemix.net/files/3_PostgreSQL_simple_dashboard.json

3. Notice that a new flow as shown in the following figure is displayed.

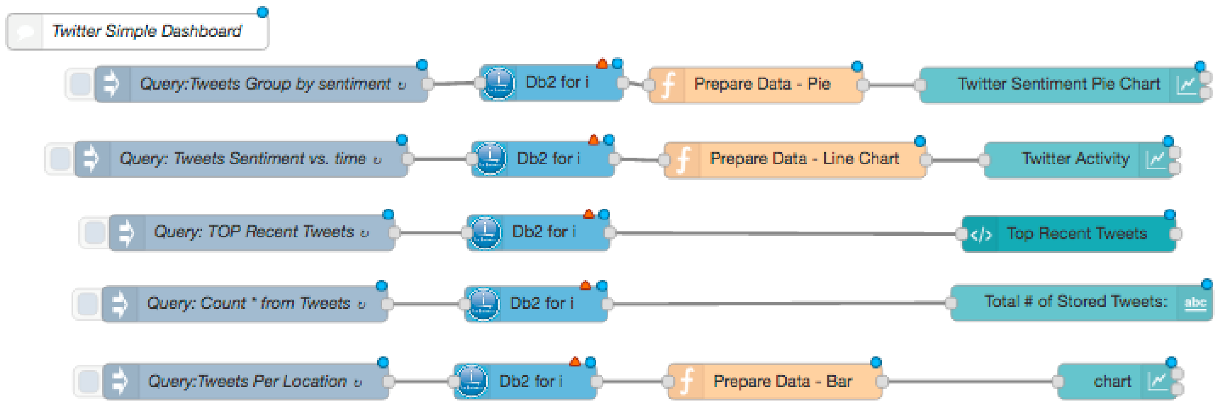

Twitter Simple Dashboard
Query:Tweets Group by sentiment
Db2 for i
Prepare Data - Pie
Twitter Sentiment Pie Chart
Query: Tweets Sentiment vs. time
Db2 for i
Prepare Data - Line Chart
Twitter Activity
Query: TOP Recent Tweets
Db2 for i
Top Recent Tweets
Query: Count * from Tweets
Db2 for i
Total # of Stored Tweets:
Query:Tweets Per Location
Db2 for i
Prepare Data - Bar
chart

Figure 20. Simple dashboard nodes with Twitter and Db2 for i

4. Configure each Db2 node with your connection information.

5. Deploy and access your dashboard. The refresh rate in the just imported code is set to 10 minutes, so if you don't want to wait, go to your flow editor and force the dashboard refresh by clicking on each inject nodes on the left side of your flow. The dashboard URL is available at the right side of the dashboard panel.

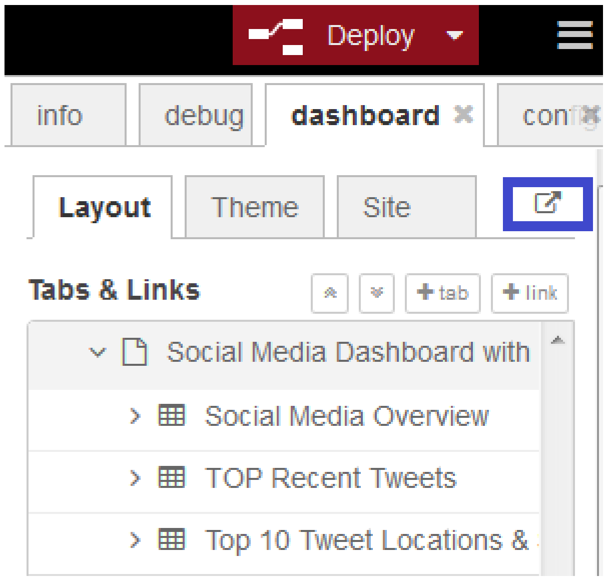
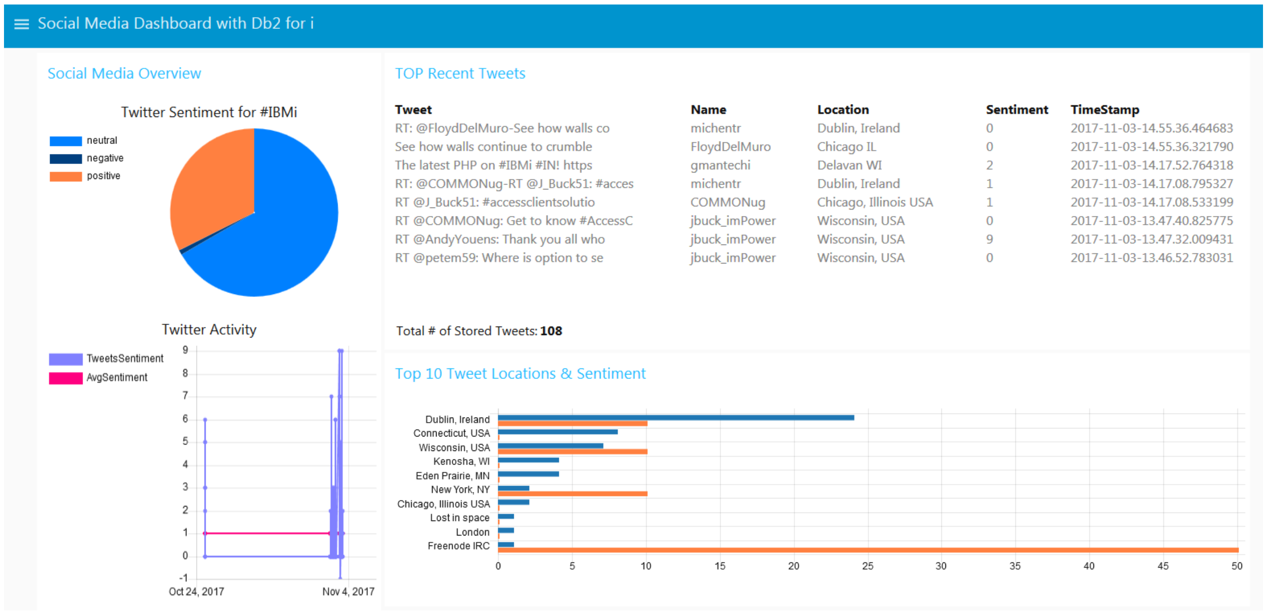


Figure 21. Node-RED dashboard layout panel and URL

You have just created a simple social media dashboard, with a few graphs populated with data residing on your Db2 for i database!



Tweet	Name	Location	Sentiment	TimeStamp
RT: @FloydDelMuro-See how walls co	michentr	Dublin, Ireland	0	2017-11-03-14.55.36.464683
See how walls continue to crumble	FloydDelMuro	Chicago IL	0	2017-11-03-14.55.36.321790
The latest PHP on #IBMi #IN! https	gmantechi	Delavan WI	2	2017-11-03-14.17.52.764318
RT: @COMMONug-RT @J_Buck51: #acces	michentr	Dublin, Ireland	1	2017-11-03-14.17.08.795327
RT @J_Buck51: #accessclientsolutio	COMMONug	Chicago, Illinois USA	1	2017-11-03-14.17.08.533199
RT @COMMONug: Get to know #AccessC	jbuck_imPower	Wisconsin, USA	0	2017-11-03-13.47.40.825775
RT @AndyYouens: Thank you all who	jbuck_imPower	Wisconsin, USA	9	2017-11-03-13.47.32.009431
RT @petem59: Where is option to se	jbuck_imPower	Wisconsin, USA	0	2017-11-03-13.46.52.783031

Total # of Stored Tweets: **108**

Figure 22: Social media dashboard on IBM i with Node-RED, Twitter APIs, and Db2 for i

Summary

"Fail fast, succeed faster"

IBM i is a private cloud on its own, and when used with open source technology in conjunction with Db2 for i, you can perform great things quickly and bring innovation for your business.

In this article, we demonstrated that Node-RED and Node.js on IBM i can be a real accelerator for API integration and prototyping, allowing you to play with nodes and program the whole world by a simple drag-and-drop action! We also had an overview of the power of IBM Watson and its cognitive computing APIs for augmenting all our solutions on IBM i.

The PostgreSQL adaptation demonstrate the same concepts and is also a good example of how to prototype using PostgreSQL on IBM Cloud/Linux/Windows or any cloud provider.

References

- Running Node-RED on IBM i: Installation and first flow: https://www.ibm.com/developerworks/ibmi/library/i-running-node-red/
- Node-RED on IBM DeveloperWorks: https://developer.ibm.com/dwblog/2016/node-red-programming-tool-open-source-iot/

- Db2 for i Node-RED node: https://flows.nodered.org/node/node-red-contrib-db2-for-i
- The powerful JSON_TABLE function: Utilize JSON information with IBM Db2 for i: https://www.ibm.com/developerworks/ibmi/library/i-json-table-trs/index.html

This content is provided as is, in the IBM Cloud & Watson day home page by the IBM Client Center, Montpellier.

http://ibmcloud-watson-day.mybluemix.net/

•