

Sprint Delivery – 3

Project	IoT Enabled Smart Farming Application
Team ID	PNT2022TMID43402
Date	10 November 2022

In the last sprint phase(sprint - 2), a node-MCU device is created in Watson IOT Platform and randomly generated .

Now, using PYTHON CODE , we need to generate random values for Temperature, Humidity and Soil moisture sensors. For that we need Python 3.7.0 IDLE.

First of all, required ibmiotf package using this command in the command prompt :

pip install ibmiotf

The required library files are included in the code ,then functions are written

CODE:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
```

#Provide your IBM Watson Device Credentials

```
organization = "75b0l2"
deviceType = "NodeMCU"
deviceId = "12345"
authMethod = "token"
authToken = "12345678"
```

Initialize GPIO

```
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status == "MotorON":
        print("motor in on")
    else :
        print ("motor is off")
```

try:

```
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method":
authMethod, "auth-token": authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)
```

```

except Exception as e:
    print("Caught exception connecting device: %s" % str(e))
    sys.exit()

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type
"greeting" 10 times
deviceCli.connect()

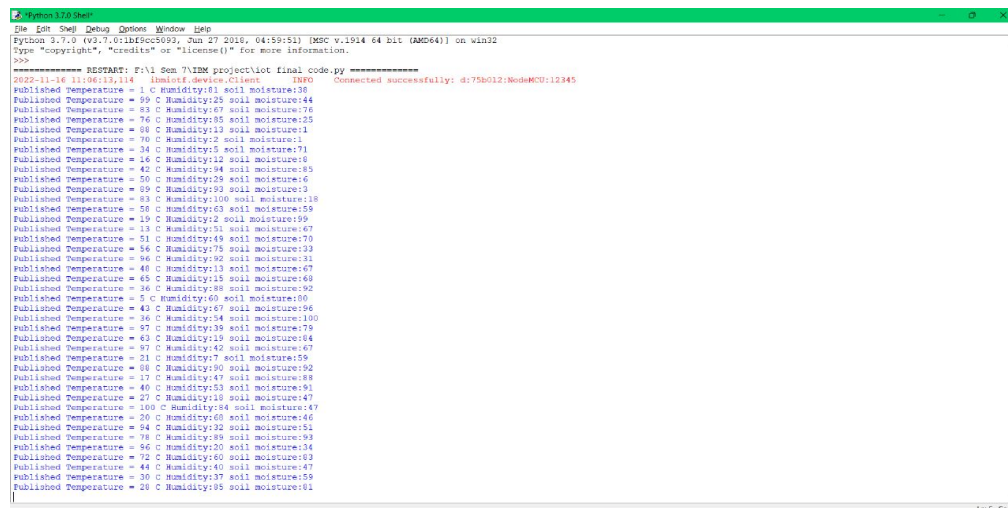
while True:
    #Get Sensor Data from DHT11
    temp=random.randint(0,100)
    humid=random.randint(0,100)
    soil=random.randint(0,100)
    data = {'temp' : temp, 'humid' : humid, 'soil' : soil}
    #print data
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % temp, "Humidity:%s" %humid, "soil
moisture:%s" %soil)

        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTTF")
            time.sleep(1)
            deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()

```

OUTPUT:

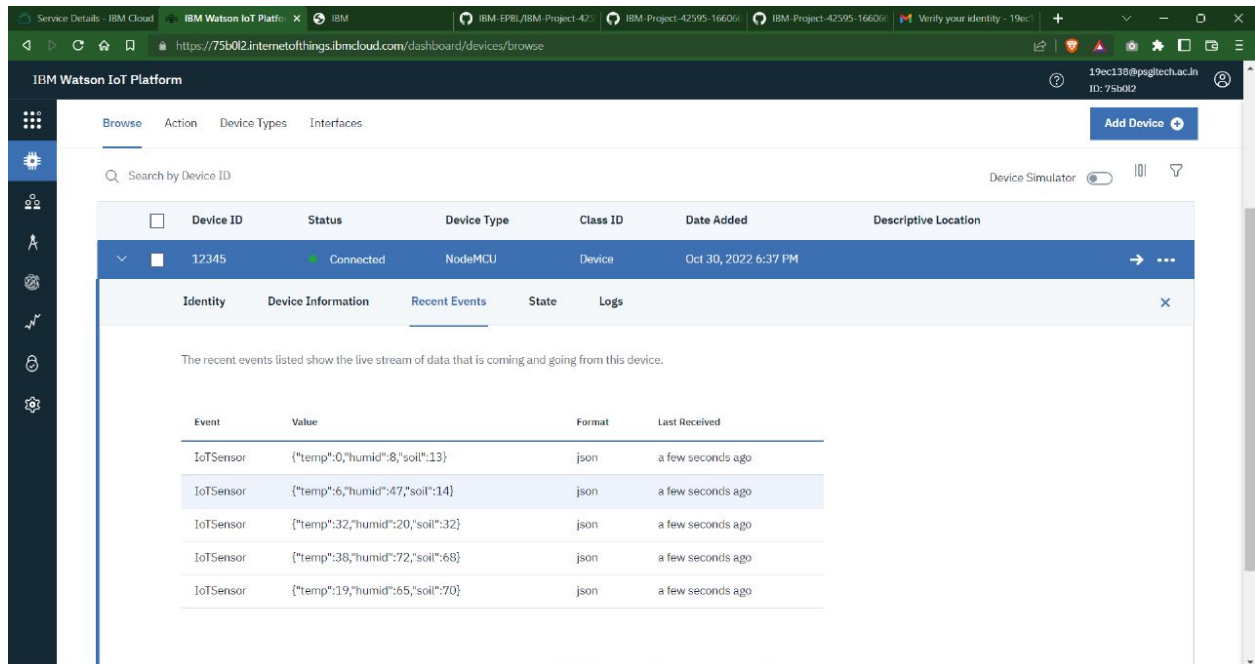


```

Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (tags/v3.7.0:1bfecf509, Jun 27 2016, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits()" or "license()" for more information.
>>>
===== RESTART: F:\1 Sem 7\IEM project\iot final code.py =====
2022-11-16 11:06:13,114 Unistart device-client 1800 Connected successfully: d:75b012:Mod#03:12345
Published Temperature = 1 C Humidity:01 soil moisture:38
Published Temperature = 99 C Humidity:25 soil moisture:44
Published Temperature = 83 C Humidity:67 soil moisture:76
Published Temperature = 76 C Humidity:95 soil moisture:25
Published Temperature = 98 C Humidity:13 soil moisture:11
Published Temperature = 70 C Humidity:2 soil moisture:11
Published Temperature = 34 C Humidity:5 soil moisture:71
Published Temperature = 16 C Humidity:15 soil moisture:8
Published Temperature = 42 C Humidity:94 soil moisture:85
Published Temperature = 50 C Humidity:28 soil moisture:6
Published Temperature = 99 C Humidity:93 soil moisture:9
Published Temperature = 83 C Humidity:100 soil moisture:18
Published Temperature = 58 C Humidity:63 soil moisture:59
Published Temperature = 19 C Humidity:2 soil moisture:99
Published Temperature = 13 C Humidity:51 soil moisture:67
Published Temperature = 51 C Humidity:49 soil moisture:75
Published Temperature = 56 C Humidity:75 soil moisture:13
Published Temperature = 96 C Humidity:92 soil moisture:31
Published Temperature = 40 C Humidity:13 soil moisture:67
Published Temperature = 65 C Humidity:15 soil moisture:60
Published Temperature = 56 C Humidity:48 soil moisture:92
Published Temperature = 5 C Humidity:60 soil moisture:80
Published Temperature = 43 C Humidity:67 soil moisture:96
Published Temperature = 36 C Humidity:54 soil moisture:180
Published Temperature = 87 C Humidity:39 soil moisture:79
Published Temperature = 63 C Humidity:19 soil moisture:84
Published Temperature = 97 C Humidity:42 soil moisture:67
Published Temperature = 21 C Humidity:7 soil moisture:56
Published Temperature = 98 C Humidity:90 soil moisture:92
Published Temperature = 17 C Humidity:47 soil moisture:88
Published Temperature = 40 C Humidity:53 soil moisture:94
Published Temperature = 27 C Humidity:18 soil moisture:47
Published Temperature = 100 C Humidity:84 soil moisture:47
Published Temperature = 20 C Humidity:60 soil moisture:46
Published Temperature = 94 C Humidity:22 soil moisture:51
Published Temperature = 78 C Humidity:89 soil moisture:93
Published Temperature = 86 C Humidity:10 soil moisture:130
Published Temperature = 72 C Humidity:60 soil moisture:63
Published Temperature = 44 C Humidity:10 soil moisture:47
Published Temperature = 30 C Humidity:37 soil moisture:59
Published Temperature = 20 C Humidity:95 soil moisture:61

```

These randomly generated values can be seen in the IBM Watson IOT Platform which is the IBM Cloud in the area of device recent events with the help of IBM **ibmiotf** package and the device credentials.

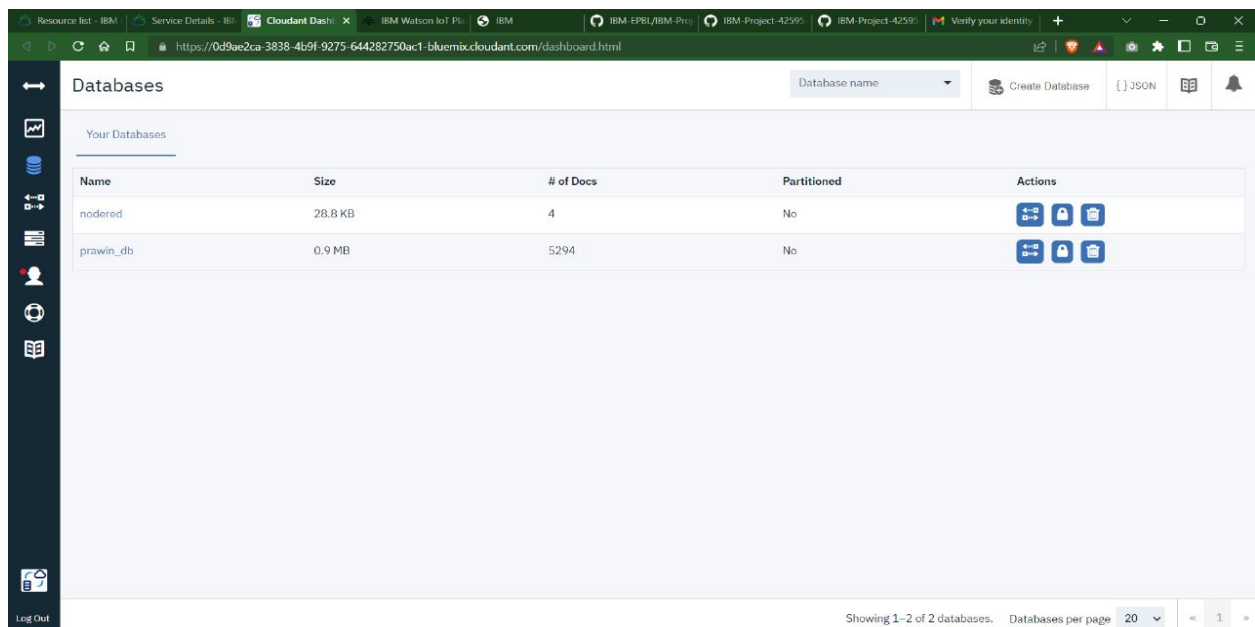


The screenshot shows the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar is present with the text 'Search by Device ID'. The main content area displays details for a device with ID '12345', which is 'Connected' and of type 'NodeMCU'. The 'Recent Events' tab is active, showing a table of sensor data. The table has columns for 'Event', 'Value', 'Format', and 'Last Received'. The data shows five events from an 'IoTSensor' with JSON-formatted values for temperature, humidity, and soil moisture, all received 'a few seconds ago'.

Event	Value	Format	Last Received
IoTSensor	{"temp":0,"humid":8,"soil":13}	json	a few seconds ago
IoTSensor	{"temp":6,"humid":47,"soil":14}	json	a few seconds ago
IoTSensor	{"temp":32,"humid":20,"soil":32}	json	a few seconds ago
IoTSensor	{"temp":38,"humid":72,"soil":68}	json	a few seconds ago
IoTSensor	{"temp":19,"humid":65,"soil":70}	json	a few seconds ago

The values are stored in the IBM Cloudant database

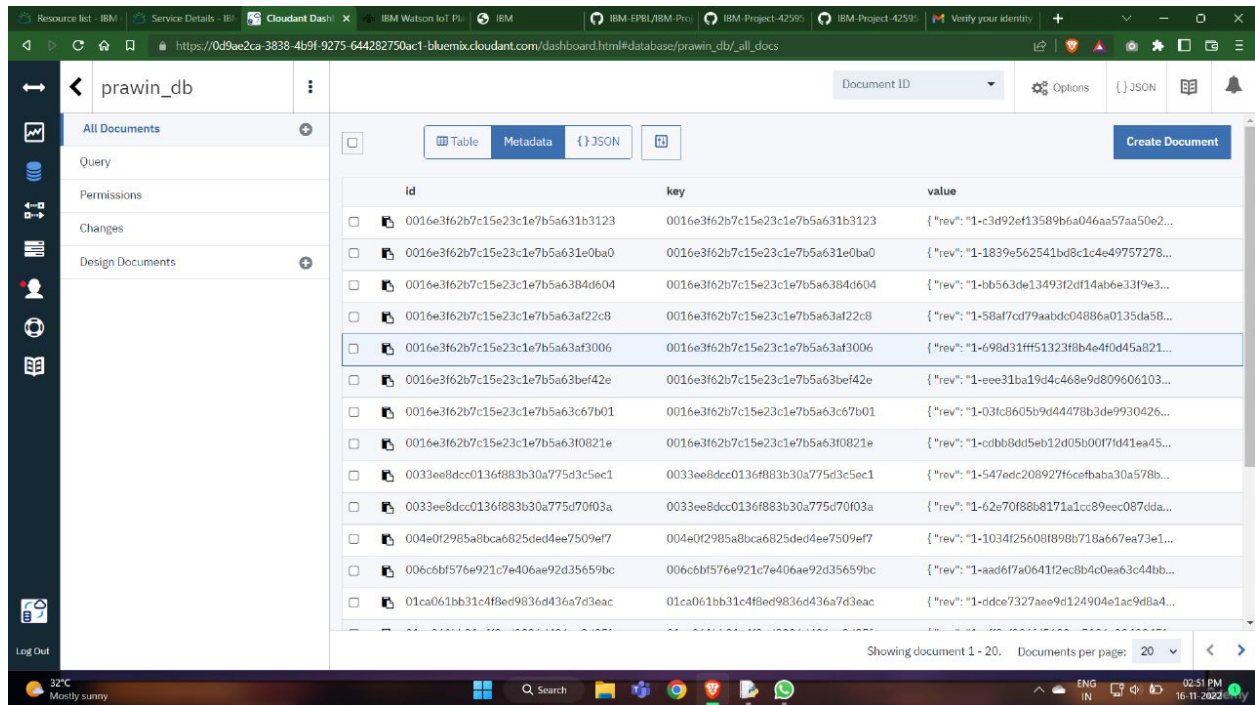
Creation of database:



The screenshot shows the IBM Cloudant Databases dashboard. The top navigation bar includes 'Database name', 'Create Database', and a JSON icon. The main content area displays a table of databases. The table has columns for 'Name', 'Size', '# of Docs', 'Partitioned', and 'Actions'. Two databases are listed: 'nodered' (28.8 KB, 4 docs, not partitioned) and 'prawin_db' (0.9 MB, 5294 docs, not partitioned). The 'Actions' column for each database contains icons for creating, deleting, and locking.

Name	Size	# of Docs	Partitioned	Actions
nodered	28.8 KB	4	No	[Create] [Delete] [Lock]
prawin_db	0.9 MB	5294	No	[Create] [Delete] [Lock]

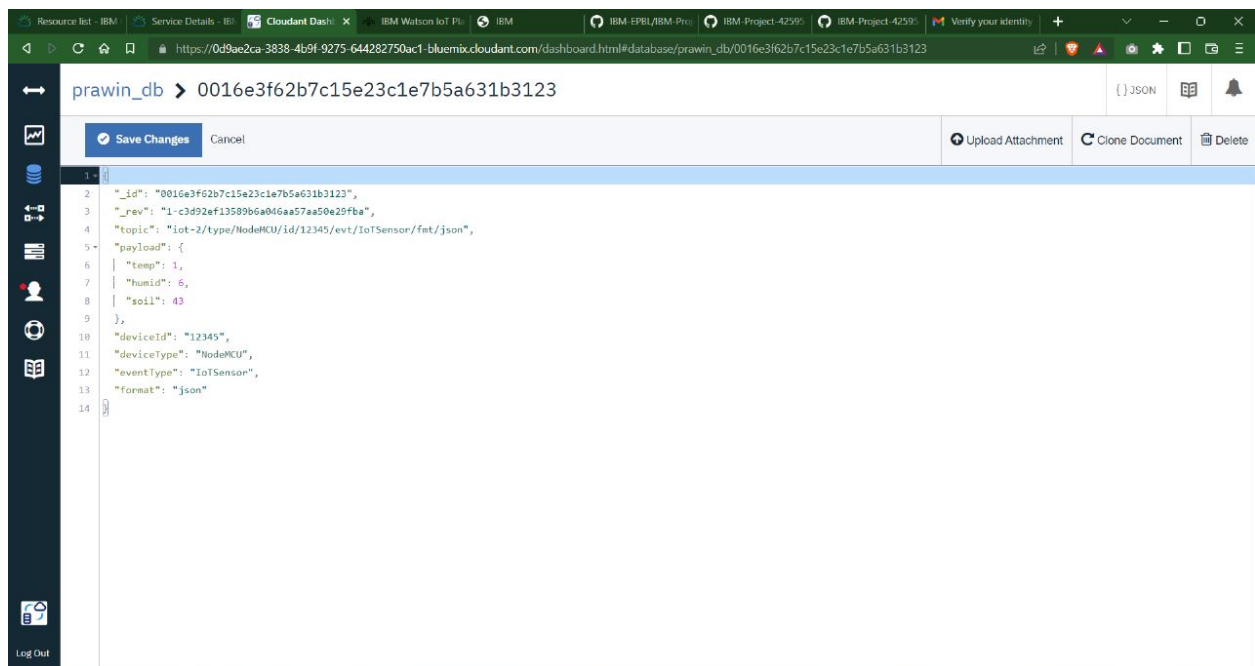
Storing data in cloud:



The screenshot shows the IBM Cloudant Dash interface for a database named 'prawin_db'. The interface includes a sidebar with navigation options like 'All Documents', 'Query', 'Permissions', 'Changes', and 'Design Documents'. The main area displays a table of documents with columns for 'id', 'key', and 'value'. The 'value' column shows JSON snippets for each document. The table is sorted by 'key' in ascending order. At the bottom, it indicates 'Showing document 1 - 20' and 'Documents per page: 20'.

id	key	value
0016e3f62b7c15e23c1e7b5a631b3123	0016e3f62b7c15e23c1e7b5a631b3123	{ "rev": "1-c3d92ef13589b6a046aa57aa50e2..." }
0016e3f62b7c15e23c1e7b5a631e0ba0	0016e3f62b7c15e23c1e7b5a631e0ba0	{ "rev": "1-1839e562541bd8c1c4e49757278..." }
0016e3f62b7c15e23c1e7b5a6384d604	0016e3f62b7c15e23c1e7b5a6384d604	{ "rev": "1-bb563de13493f2df14ab6e33f9e3..." }
0016e3f62b7c15e23c1e7b5a63af22c8	0016e3f62b7c15e23c1e7b5a63af22c8	{ "rev": "1-58af7cd79aabcd04886a0135da58..." }
0016e3f62b7c15e23c1e7b5a63af3006	0016e3f62b7c15e23c1e7b5a63af3006	{ "rev": "1-698d31ff51323f8b4e4f0d45a821..." }
0016e3f62b7c15e23c1e7b5a63bef42e	0016e3f62b7c15e23c1e7b5a63bef42e	{ "rev": "1-eee31ba19d4c468e9d809606103..." }
0016e3f62b7c15e23c1e7b5a63c67b01	0016e3f62b7c15e23c1e7b5a63c67b01	{ "rev": "1-03fc8605b9d4478b3de9930426..." }
0016e3f62b7c15e23c1e7b5a63f0821e	0016e3f62b7c15e23c1e7b5a63f0821e	{ "rev": "1-cdbb8dd5eb12d05b00f7d41ea45..." }
0033ee8dcc0136f883b30a775d3c5ec1	0033ee8dcc0136f883b30a775d3c5ec1	{ "rev": "1-547edc208927f6cefbaba30a578b..." }
0033ee8dcc0136f883b30a775d70f03a	0033ee8dcc0136f883b30a775d70f03a	{ "rev": "1-62e70f88b8171a1cc89e087dda..." }
004e0f2985a8bca6825ded4ee7509ef7	004e0f2985a8bca6825ded4ee7509ef7	{ "rev": "1-1034f25608f898b718a667ea73e1..." }
006c6bf576e921c7e406ae92d35659bc	006c6bf576e921c7e406ae92d35659bc	{ "rev": "1-aad6f7a0641f2ec8b4c0ea63c44bb..." }
01ca061bb31c4f8ed9836d436a7d3eac	01ca061bb31c4f8ed9836d436a7d3eac	{ "rev": "1-dcce7327aee9d124904e1ac9d8a4..." }

Actual Json format stored in cloud:



The screenshot shows the IBM Cloudant Dash interface for a specific document in the 'prawin_db' database. The document ID is '0016e3f62b7c15e23c1e7b5a631b3123'. The document is displayed in JSON format. The JSON structure includes fields for '_id', '_rev', 'topic', 'payload', 'deviceId', 'deviceType', 'eventType', and 'format'.

```
1 {
2   "_id": "0016e3f62b7c15e23c1e7b5a631b3123",
3   "_rev": "1-c3d92ef13589b6a046aa57aa50e29fba",
4   "topic": "iot-2/type/NodeMCU/id/12345/evt/IoTSensor/fmt/json",
5   "payload": {
6     "temp": 1,
7     "humid": 6,
8     "soil": 43
9   },
10  "deviceId": "12345",
11  "deviceType": "NodeMCU",
12  "eventType": "IoTSensor",
13  "format": "json"
14 }
```