

TEAM MANAGEMENT FOR AGILE PLANNING

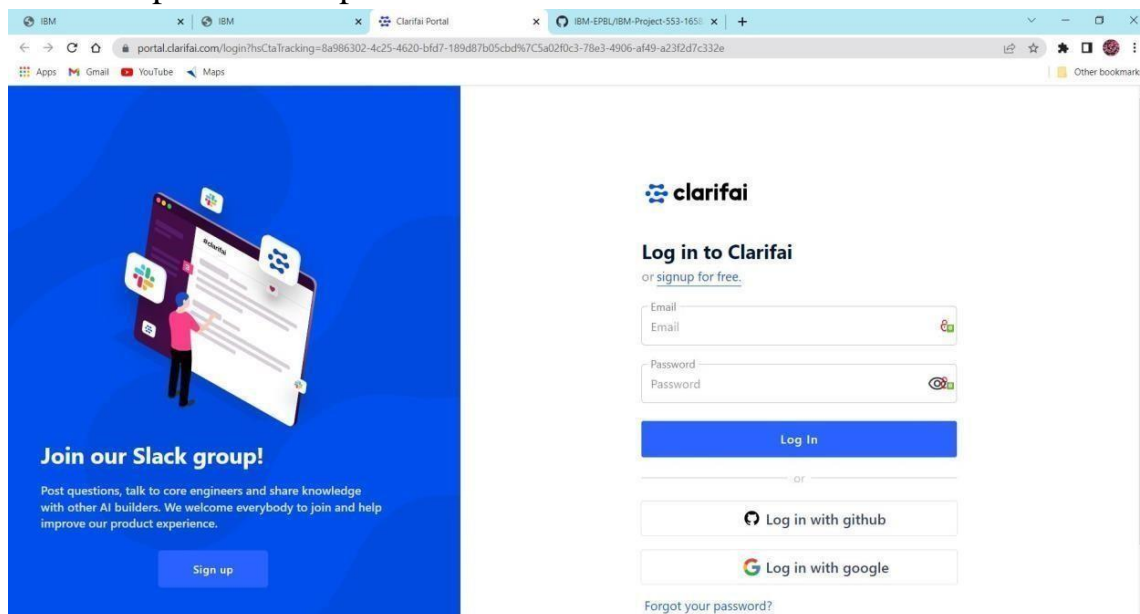
TEAM ID : PNT2022TMID17378

CLARIFAI:

Clarifai provides an end-to-end platform with the easiest to use UI and API in the market. Clarifai Inc. is an artificial intelligence (AI) company that specializes in computer vision and uses machine learning and deep neural networks to identify and analyse images and videos. The company offers its solution via API, mobile SDK, and on-premise solutions.

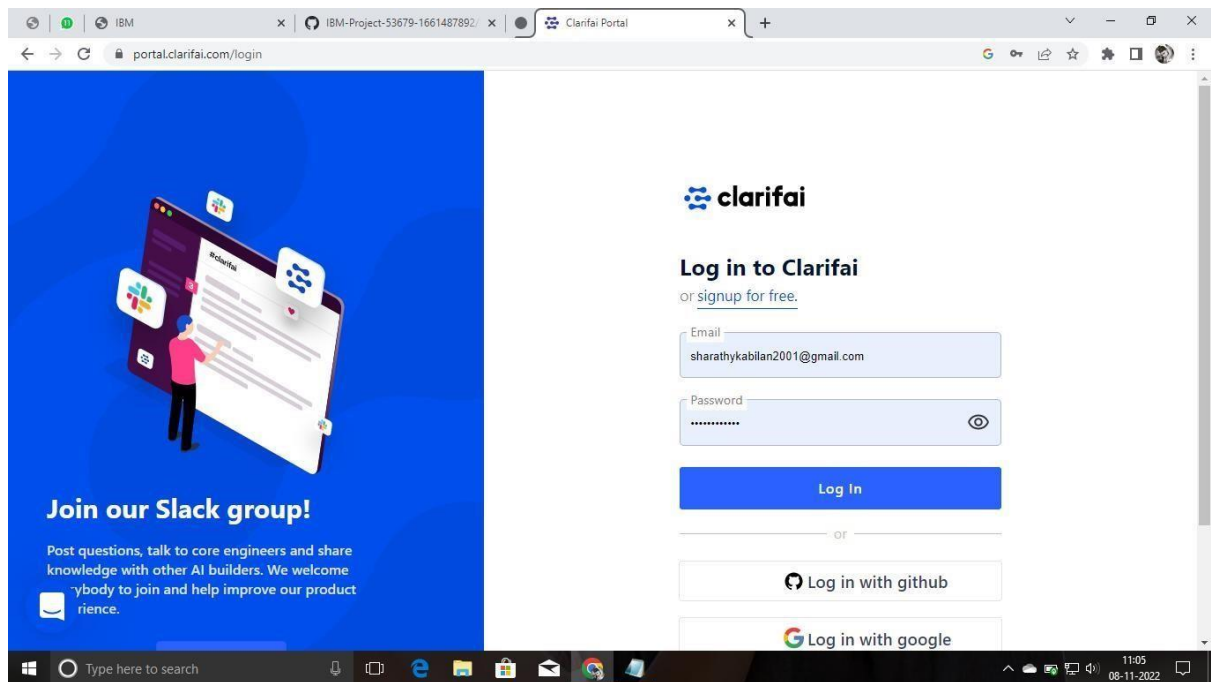
STEP 1:

- Open Clarifai portal in web browser.



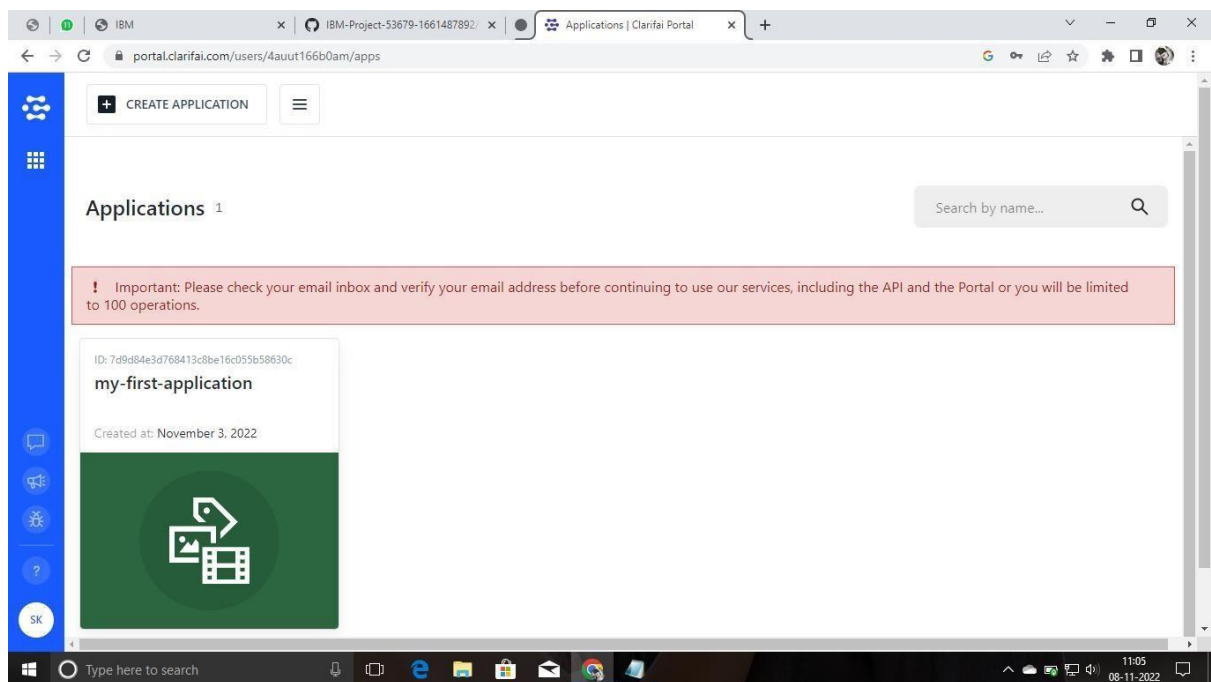
STEP 2:

- Signup using the required user mail and password



STEP 3:

Finally, Created an account



IBM WATSON PLATFORM:

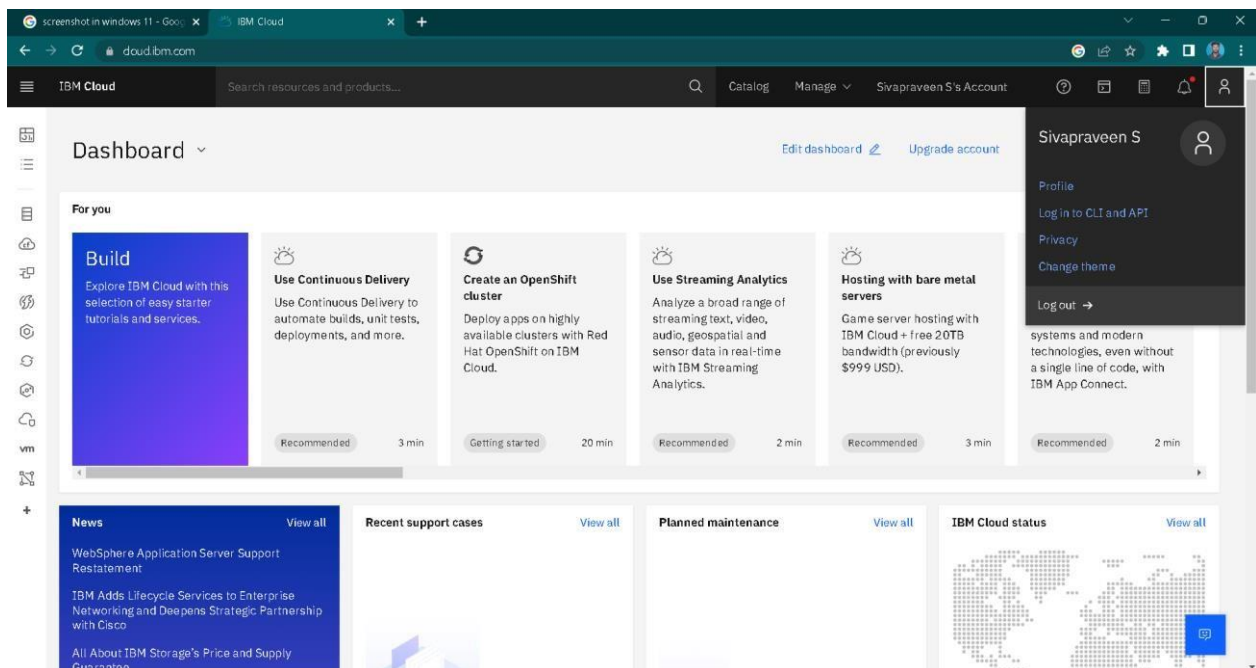
We need to have basic knowledge of the following cloud services:

- IBM Watson IoT Platform

- Node-RED Service
- Cloudbant DB

We need to create an IBM Cloud Account to complete this project.

LOGIN:

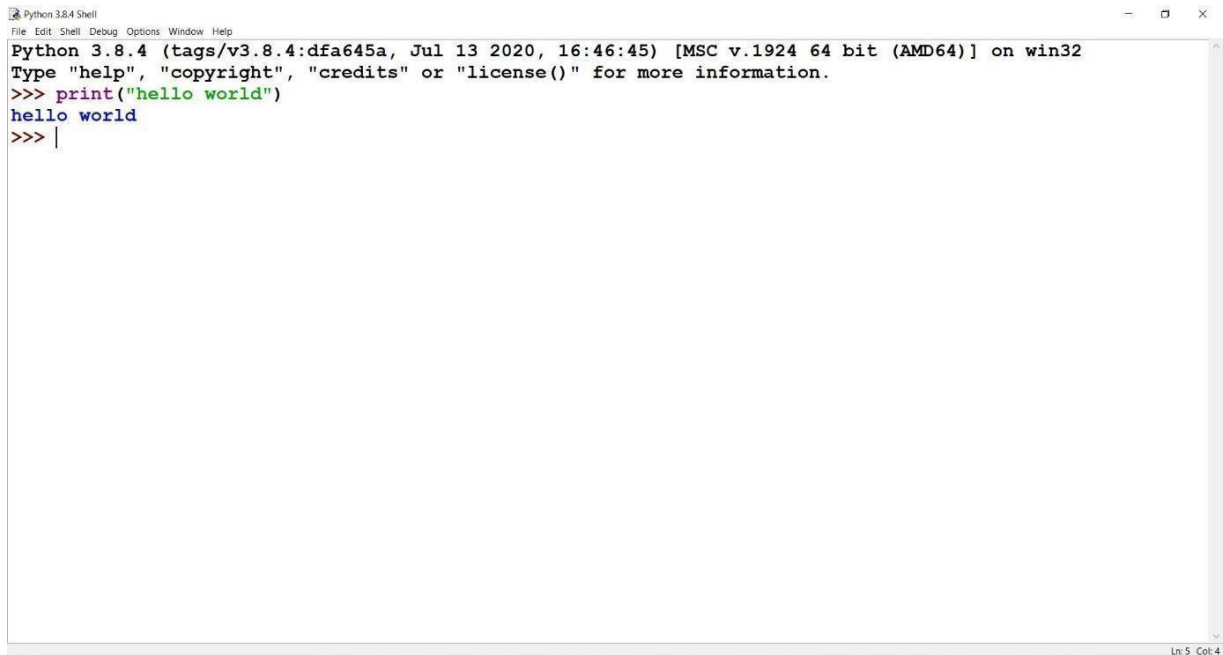


PYTHON IDLE INSTALLATION:

Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a generalpurpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems.

STEP 1:

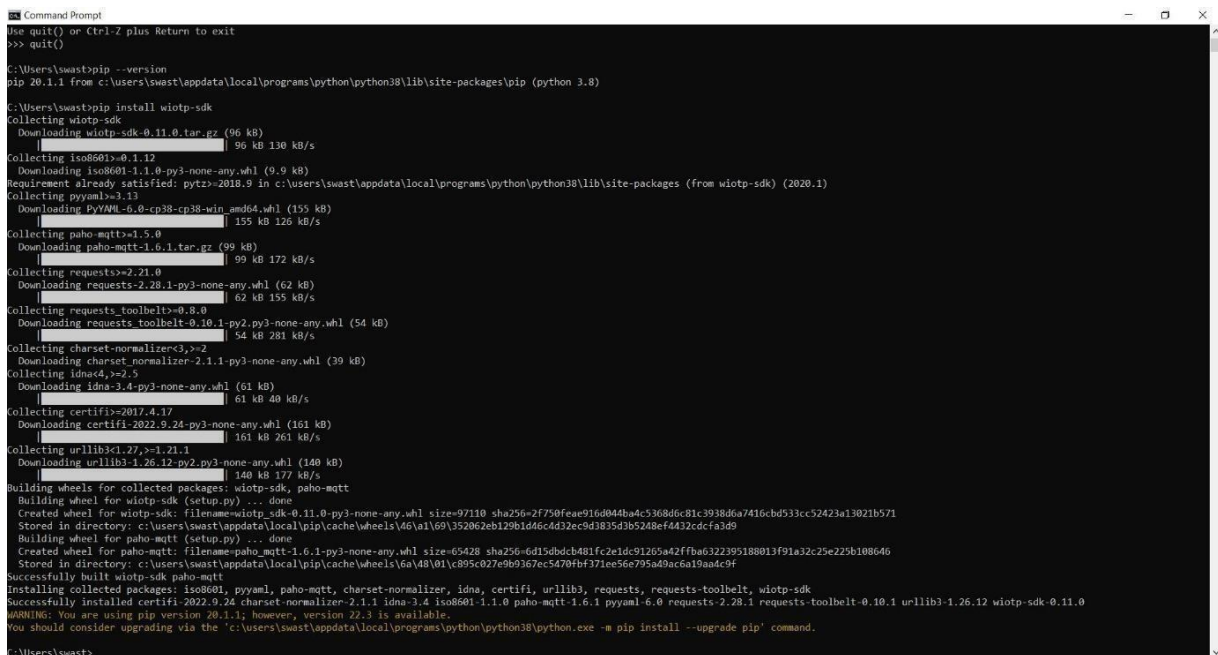
- Python is installed successfully



```
Python 3.8.4 Shell
File Edit Shell Debug Options Window Help
Python 3.8.4 (tags/v3.8.4:dfa645a, Jul 13 2020, 16:46:45) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("hello world")
hello world
>>> |
```

STEP 2:

- The required python libraries are installed.
- Watson Python SDK to connect to IBM Watson Platform using python code is installed
- pip install wiotp-sdk



```
Command Prompt
use quit() or Ctrl-Z plus Return to exit
>>> quit()

C:\Users\swast>pip --version
pip 20.1.1 from c:\users\swast\appdata\local\programs\python\python38\lib\site-packages\pip (python 3.8)

C:\Users\swast>pip install wiotp-sdk
Collecting wiotp-sdk
  Downloading wiotp-sdk-0.11.0.tar.gz (96 kB)
    |#####| 96 kB 130 kB/s
Collecting iso8601>=0.1.12
  Downloading iso8601-1.1.0-py3-none-any.whl (9.9 kB)
Requirement already satisfied: pytz>=2018.9 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from wiotp-sdk) (2020.1)
Collecting pyyaml>=3.13
  Downloading PyYAML-6.0-cp38-cp38-win_amd64.whl (155 kB)
    |#####| 155 kB 126 kB/s
Collecting paho-mqtt>=1.5.0
  Downloading paho-mqtt-1.6.1.tar.gz (99 kB)
    |#####| 99 kB 172 kB/s
Collecting requests>=2.21.0
  Downloading requests-2.28.1-py3-none-any.whl (62 kB)
    |#####| 62 kB 155 kB/s
Collecting requests-toolbelt>=0.8.0
  Downloading requests-toolbelt-0.10.1-py2.py3-none-any.whl (54 kB)
    |#####| 54 kB 281 kB/s
Collecting charset-normalizer<3,>=2
  Downloading charset-normalizer-2.1.1-py3-none-any.whl (39 kB)
Collecting idna<4,>=2.5
  Downloading idna-3.4-py3-none-any.whl (61 kB)
    |#####| 61 kB 40 kB/s
Collecting certifi>=2017.4.17
  Downloading certifi-2022.9.24-py3-none-any.whl (161 kB)
    |#####| 161 kB 261 kB/s
Collecting urllib3<1.27,>=1.21.1
  Downloading urllib3-1.26.12-py2.py3-none-any.whl (140 kB)
    |#####| 140 kB 177 kB/s
Building wheels for collected packages: wiotp-sdk, paho-mqtt
  Building wheel for wiotp-sdk (setup.py) ... done
    Created wheel for wiotp-sdk: filename=wiotp_sdk-0.11.0-py3-none-any.whl size=97110 sha256=2f750feae916d044ba4c5368d6c81c3938d6a7416cbd533cc52423a13021b571
    Stored in directory: c:\users\swast\appdata\local\pip\cache\wheels\46\al\69\352062eb129b1d46c4d32ec9d3835d3b5248ef4432cdcf3d9
  Building wheel for paho-mqtt (setup.py) ... done
    Created wheel for paho-mqtt: filename=paho_mqtt-1.6.1-py3-none-any.whl size=65428 sha256=6d15dbdcb481fc2e1dc91265a42ffba6322395188013f91a32c25e225b108646
    Stored in directory: c:\users\swast\appdata\local\pip\cache\wheels\6a\48\01\c895c027e9b9367ec5470fbf371ee56e795a49ac6a19aadcf9f
Successfully built wiotp-sdk paho-mqtt
Installing collected packages: iso8601, pyyaml, paho-mqtt, charset-normalizer, idna, certifi, urllib3, requests, requests-toolbelt, wiotp-sdk
Successfully installed certifi-2022.9.24 charset-normalizer-2.1.1 idna-3.4 iso8601-1.1.0 paho-mqtt-1.6.1 pyyaml-6.0 requests-2.28.1 requests-toolbelt-0.10.1 urllib3-1.26.12 wiotp-sdk-0.11.0
WARNING: You are using pip version 20.1.1; however, version 22.3 is available.
You should consider upgrading via the 'c:\users\swast\appdata\local\programs\python\python38\python.exe -m pip install --upgrade pip' command.

C:\Users\swast>
```

- Python client library for IBM Text to Speech is installed
- `pip install --upgrade "ibm-watson>=5.0.0"`

```

Command Prompt
C:\Users\swast>pip install --upgrade "ibm-watson>=5.0.0"
Collecting ibm-watson<=5.0.0
  Downloading ibm-watson-6.1.0.tar.gz (373 kB)
    | 83 kB 152 kB/s
    Installing build dependencies ... done
    Getting requirements to build wheel ... done
    Preparing wheel metadata ... done
Collecting ibm-cloud-sdk-core==3.*>=3.3.6
  Downloading ibm-cloud-sdk-core-3.16.0-py3-none-any.whl (83 kB)
    | 68 kB 195 kB/s
Requirement already satisfied, skipping upgrade: requests<3.0,>=2.0 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from ibm-watson<=5.0.0) (2.28.1)
Collecting websocket-client==1.1.0
  Downloading websocket-client-1.1.0-py2.py3-none-any.whl (68 kB)
    | 68 kB 195 kB/s
Requirement already satisfied, skipping upgrade: python-dateutil>=2.5.3 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from ibm-watson<=5.0.0) (2.8.1)
Collecting PyJWT<3.0.0,>=2.4.0
  Downloading PyJWT-2.6.0-py3-none-any.whl (20 kB)
    | 20 kB 195 kB/s
Requirement already satisfied, skipping upgrade: urllib3<2.0.0,>=1.26.0 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from ibm-cloud-sdk-core==3.*>=3.3.6->ibm-watson<=5.0.0) (1.26.0)
Requirement already satisfied, skipping upgrade: certifi>=2017.4.17 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from requests<3.0,>=2.0->ibm-watson<=5.0.0) (2022.9.24)
Requirement already satisfied, skipping upgrade: idna<4,>=2.5 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from requests<3.0,>=2.0->ibm-watson<=5.0.0) (3.4)
Requirement already satisfied, skipping upgrade: charset-normalizer<3,>=2 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from requests<3.0,>=2.0->ibm-watson<=5.0.0) (2.1.1)
Requirement already satisfied, skipping upgrade: six>=1.5 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from python-dateutil>=2.5.3->ibm-watson<=5.0.0) (1.15.0)
Building wheels for collected packages: ibm-watson
  Building wheel for ibm-watson (PEP 517) ... done
  Created wheel for ibm-watson: filename=ibm_watson-6.1.0-py3-none-any.whl size=370748 sha256=50648b8c1c54ee0ba24e5cc521f68536cb77a9cf975f5cc5f975bddf9ba6956
  Stored in directory: c:\users\swast\appdata\local\pip\cache\wheels\34\b4\cd\829a351c802b7a578115fe7ddaedff62b29eae84e90882c7e2
Successfully built ibm-watson
Installing collected packages: PyJWT, ibm-cloud-sdk-core, websocket-client, ibm-watson
Successfully installed PyJWT-2.6.0 ibm-cloud-sdk-core-3.16.0 ibm-watson-6.1.0 websocket-client-1.1.0
WARNING: You are using pip version 20.1.1; however, version 22.3 is available.
You should consider upgrading via the 'c:\users\swast\appdata\local\programs\python\python38\python.exe -m pip install --upgrade pip' command.

C:\Users\swast>

```

- Required Libraries for cloud object storage is installed.
- `pip install ibm-cos-sdk`

```

Command Prompt
C:\Users\swast>pip install ibm-cos-sdk
Collecting ibm-cos-sdk
  Downloading ibm-cos-sdk-2.12.0.tar.gz (55 kB)
    | 55 kB 411 kB/s
Collecting ibm-cos-sdk-core==2.12.0
  Downloading ibm-cos-sdk-core-2.12.0.tar.gz (956 kB)
    | 956 kB 251 kB/s
Collecting ibm-cos-sdk-s3transfer==2.12.0
  Downloading ibm-cos-sdk-s3transfer-2.12.0.tar.gz (135 kB)
    | 135 kB 242 kB/s
Collecting jmespath<1.0.0,>=0.10.0
  Downloading jmespath-0.10.0-py2.py3-none-any.whl (24 kB)
    | 24 kB 195 kB/s
Collecting python-dateutil<3.0.0,>=2.8.2
  Downloading python-dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
    | 247 kB 261 kB/s
Requirement already satisfied: requests<3.0,>=2.27.1 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from ibm-cos-sdk-core==2.12.0->ibm-cos-sdk) (2.28.1)
Requirement already satisfied: urllib3<3.0,>=1.26.9 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from ibm-cos-sdk-core==2.12.0->ibm-cos-sdk) (1.26.12)
Requirement already satisfied: six>=1.5 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from python-dateutil<3.0.0,>=2.8.2->ibm-cos-sdk-core==2.12.0->ibm-cos-sdk) (1.15.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from requests<3.0,>=2.27.1->ibm-cos-sdk-core==2.12.0->ibm-cos-sdk) (2022.9.24)
Requirement already satisfied: idna<4,>=2.5 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from requests<3.0,>=2.27.1->ibm-cos-sdk-core==2.12.0->ibm-cos-sdk) (3.4)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from requests<3.0,>=2.27.1->ibm-cos-sdk-core==2.12.0->ibm-cos-sdk) (2.1.1)
Building wheels for collected packages: ibm-cos-sdk, ibm-cos-sdk-core, ibm-cos-sdk-s3transfer
  Building wheel for ibm-cos-sdk (setup.py) ... done
  Created wheel for ibm-cos-sdk: filename=ibm_cos_sdk-2.12.0-py3-none-any.whl size=73926 sha256=a6f05ca0730b69209e285e7f0e185c5bfa4721a71f535188f94c734e01cd3be
  Stored in directory: c:\users\swast\appdata\local\pip\cache\wheels\21\5f\fd\6a04fb45aad71bc0c8300834368f9d39ef7c4fd1869d2244d
  Building wheel for ibm-cos-sdk-core (setup.py) ... done
  Created wheel for ibm-cos-sdk-core: filename=ibm_cos_sdk_core-2.12.0-py3-none-any.whl size=562952 sha256=c7f8e80ee7f511d484073c508253371db715bad59fb3deda4ad0d38a3f99d7
  Stored in directory: c:\users\swast\appdata\local\pip\cache\wheels\ca\3d\78\48c57e974d77098f3facc81d4f402b1d6ba8cbf8c69d660ee7
  Building wheel for ibm-cos-sdk-s3transfer (setup.py) ... done
  Created wheel for ibm-cos-sdk-s3transfer: filename=ibm_cos_sdk_s3transfer-2.12.0-py3-none-any.whl size=89769 sha256=67c5983a6db0be33db07cbcd33d7216ebef83fec9e5f0275d9fe8e51ceb77
  Stored in directory: c:\users\swast\appdata\local\pip\cache\wheels\c0\7a\37\13b53ca7027a29a1062a47c58baa1c4f3832795b698c6bd46
Successfully built ibm-cos-sdk ibm-cos-sdk-core ibm-cos-sdk-s3transfer
Installing collected packages: jmespath, python-dateutil, ibm-cos-sdk-core, ibm-cos-sdk-s3transfer, ibm-cos-sdk
Attempting uninstall: python-dateutil
  Found existing installation: python-dateutil 2.8.1
  Successfully uninstalled python-dateutil-2.8.1
Successfully installed ibm-cos-sdk-2.12.0 ibm-cos-sdk-core-2.12.0 ibm-cos-sdk-s3transfer-2.12.0 jmespath-0.10.0 python-dateutil-2.8.2
WARNING: You are using pip version 20.1.1; however, version 22.3 is available.
You should consider upgrading via the 'c:\users\swast\appdata\local\programs\python\python38\python.exe -m pip install --upgrade pip' command.

C:\Users\swast>

```

- `pip install -U ibm-cos-sdk`

```
Command Prompt
WARNING: You are using pip version 20.1.1; however, version 22.3 is available.
You should consider upgrading via the 'c:\users\swast\appdata\local\programs\python\python38\python.exe -m pip install --upgrade pip' command.

C:\Users\swast>pip install -U ibm-cos-sdk
Requirement already up-to-date: ibm-cos-sdk in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (2.12.0)
Requirement already satisfied, skipping upgrade: ibm-cos-sdk-s3transfer==2.12.0 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from ibm-cos-sdk) (2.12.0)
Requirement already satisfied, skipping upgrade: ibm-cos-sdk-core==2.12.0 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from ibm-cos-sdk) (2.12.0)
Requirement already satisfied, skipping upgrade: jmespath<1.0.0,>=0.10.0 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from ibm-cos-sdk) (0.10.0)
Requirement already satisfied, skipping upgrade: requests<3.0,>=2.27.1 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from ibm-cos-sdk-core==2.12.0->ibm-cos-sdk) (2.28.1)
Requirement already satisfied, skipping upgrade: urllib3<1.27,>=1.26.9 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from ibm-cos-sdk-core==2.12.0->ibm-cos-sdk) (1.26.12)
Requirement already satisfied, skipping upgrade: python-dateutil<3.0.0,>=2.8.2 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from ibm-cos-sdk-core==2.12.0->ibm-cos-sdk) (2.8.2)
Requirement already satisfied, skipping upgrade: charset-normalizer<3,>=2 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from requests<3.0,>=2.27.1->ibm-cos-sdk-core==2.12.0->ibm-cos-sdk) (2.1.1)
Requirement already satisfied, skipping upgrade: idna<4,>=2.5 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from requests<3.0,>=2.27.1->ibm-cos-sdk-core==2.12.0->ibm-cos-sdk) (3.4)
Requirement already satisfied, skipping upgrade: certifi>2017.4.17 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from requests<3.0,>=2.27.1->ibm-cos-sdk-core==2.12.0->ibm-cos-sdk) (2022.9.24)
Requirement already satisfied, skipping upgrade: six>=1.5 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from python-dateutil<3.0.0,>=2.8.2->ibm-cos-sdk-core==2.12.0->ibm-cos-sdk) (1.15.0)
WARNING: You are using pip version 20.1.1; however, version 22.3 is available.
You should consider upgrading via the 'c:\users\swast\appdata\local\programs\python\python38\python.exe -m pip install --upgrade pip' command.

C:\Users\swast>
```

- pip install boto3

```
Command Prompt
WARNING: You are using pip version 20.1.1; however, version 22.3 is available.
You should consider upgrading via the 'c:\users\swast\appdata\local\programs\python\python38\python.exe -m pip install --upgrade pip' command.

C:\Users\swast>pip install boto3
Collecting boto3
  Downloading boto3-1.26.0-py3-none-any.whl (132 kB)
    |#####| 132 kB 148 kB/s
Collecting s3transfer<0.7.0,>=0.6.0
  Downloading s3transfer-0.6.0-py3-none-any.whl (79 kB)
    |#####| 79 kB 113 kB/s
Collecting botocore<1.30.0,>=1.29.0
  Downloading botocore-1.29.0-py3-none-any.whl (9.8 MB)
    |#####| 9.8 MB 2.2 MB/s
Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from boto3) (0.10.0)
Requirement already satisfied: urllib3<1.27,>=1.25.4 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from botocore<1.30.0,>=1.29.0->boto3) (1.26.12)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from botocore<1.30.0,>=1.29.0->boto3) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from python-dateutil<3.0.0,>=2.1->botocore<1.30.0,>=1.29.0->boto3) (1.15.0)
Installing collected packages: botocore, s3transfer, boto3
Successfully installed boto3-1.26.0 botocore-1.29.0 s3transfer-0.6.0
WARNING: You are using pip version 20.1.1; however, version 22.3 is available.
You should consider upgrading via the 'c:\users\swast\appdata\local\programs\python\python38\python.exe -m pip install --upgrade pip' command.

C:\Users\swast>
```

- pip install resources


```
Command Prompt
C:\Users\swast>pip install resources
Collecting resources
  Downloading resources-0.0.1.tar.gz (3.7 kB)
Building wheels for collected packages: resources
  Building wheel for resources (setup.py) ... done
  Created wheel for resources: filename=resources-0.0.1-py3-none-any.whl size=4170 sha256=38111eb3ac96cbf54f0f22303a68aee6aaca970211e26ae94f9b2441ec318e
  Stored in directory: c:\users\swast\appdata\local\pip\cache\wheels\b3\1d\00\45ae97c7b92d145a0963f711c6d22f9af5306e74c88f2f28fd
Successfully built resources
Installing collected packages: resources
Successfully installed resources-0.0.1
WARNING: You are using pip version 20.1.1; however, version 22.3 is available.
You should consider upgrading via the 'c:\users\swast\appdata\local\programs\python\python38\python.exe -m pip install --upgrade pip' command.
C:\Users\swast>
```

- pip install cloudant

```
Command Prompt
You should consider upgrading via the 'c:\users\swast\appdata\local\programs\python\python38\python.exe -m pip install --upgrade pip' command.
C:\Users\swast>pip install cloudant
Collecting cloudant
  Downloading cloudant-2.15.0-py3-none-any.whl (80 kB)
    80 kB 395 kB/s
Requirement already satisfied: requests<3.0.0,>=2.7.0 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from cloudant) (2.28.1)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from requests<3.0.0,>=2.7.0->cloudant) (2.1.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from requests<3.0.0,>=2.7.0->cloudant) (1.26.12)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from requests<3.0.0,>=2.7.0->cloudant) (2022.9.24)
Requirement already satisfied: idna<4,>=2.5 in c:\users\swast\appdata\local\programs\python\python38\lib\site-packages (from requests<3.0.0,>=2.7.0->cloudant) (3.4)
Installing collected packages: cloudant
Successfully installed cloudant-2.15.0
WARNING: You are using pip version 20.1.1; however, version 22.3 is available.
You should consider upgrading via the 'c:\users\swast\appdata\local\programs\python\python38\python.exe -m pip install --upgrade pip' command.
C:\Users\swast>
```

FROM PYTHON TO IBM:

Python code to generate random data and pass it to IBM Watson IoT

platform **SourceCode:** import time import sys import

ibmiotf.application import

ibmiotf.device import

random

#Provide your IBM Watson Device

Credentialsorganization = "wu5b55" deviceType

= "crop1" deviceId =

"1234" authMethod =

"token" authToken =

"1234567890"

Initialize GPIOtry:

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.conn

ect()while True:

temp=random.randint(0,


```
100)
Hum=random.randint(0,1
00)
moisture=random.randint
(0,100)
```

```
data = { 'temperature' : temp, 'Humidity': Hum, 'Moisture':moisture }
```

```
def myOnPublishCallback():
```

```
    print ("Temperature = " + str(temp)+" C Humidity = " +
str(hum)+ " moisture = " +str(moisture) + "to IBM Watson")
```

```
    success = deviceCli.publishEvent("IoTSensor",
"json", data, qos=0,on_publish=myOnPublishCallback)
```

```
    if not success:
```

```
        print("Not connected to
IoT")time.sleep(10)
```

```
deviceCli.commandCallback = myCommandCallback
```

```
# Disconnect the device and application from
the clouddeviceCli.disconnect()
```

DATA GENERATION PLATFORM:

Source code is deployed on IBM Watson IoT platform to generate sensor data.

SourceCode:

```
{  
  
  "temperature": random(0, 100),  
  
  "humidity": random(0, 100),  
  
  "moisture": random(0, 100),  
  
  "animalDetected":random(0,2)  
  
}
```

Output:

The screenshot displays the IBM Watson IoT Platform interface. The main panel shows a table of recent events for a device named 'cropProtection'. The table has two columns: 'Event' and 'Value'. The 'Value' column contains JSON payloads with sensor data.

Event	Value
event_1	{"temp":93,"hum":16,"moisture":97,"anim
event_1	{"temp":90,"hum":73,"moisture":15,"anim
event_1	{"temp":77,"hum":86,"moisture":87,"anim

On the right, a configuration panel for 'Device Type: crop' is visible. It shows the 'Event type name' as 'event_1' and the 'Schedule' as 'Every Minute'. The 'Payload' section displays the JSON code used for data generation:

```
{  
  0 {  
    1 "temp": random(0, 100),  
    2 "hum": random(0, 100),  
    3 "moisture": random(0, 100),  
    4 "animalDetected": random(0, 2),  
    5 }  
  6 }
```

PYTHON CODE TO IBM:

```
import time
import sys

import ibmiotf.application import
ibmiotf.device import random


#Provide your IBM Watson Device Credentials organization
= "wu5b55" deviceType =
"crop1" deviceId = "1234" authMethod = "token"
authToken = "1234567890"


# Initialize GPIO


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 thecloud as
an event of type "greeting" 10 times
deviceCli.connect() while
True:

    #Get Sensor Data from DHT11
```

```
temp=random.randint(0,100)
Hum=random.randint(0,100)
moisture=random.randint(0,100)
```

```
data = { 'temperature' : temp, 'Humidity': Hum,
'Moisture':moisture }
```

```
#print data
def myOnPublishCallback():
```

```
    print ("Temperature = " + str(temp)+" C Humidity = " +
str(hum)+ " moisture = " + str(moisture) + "to IBM Watson")
```

```
    success = deviceCli.publishEvent("IoTSensor", "json", data,qos=0,
on_publish=myOnPublishCallback) if
```

```
    not success:
```

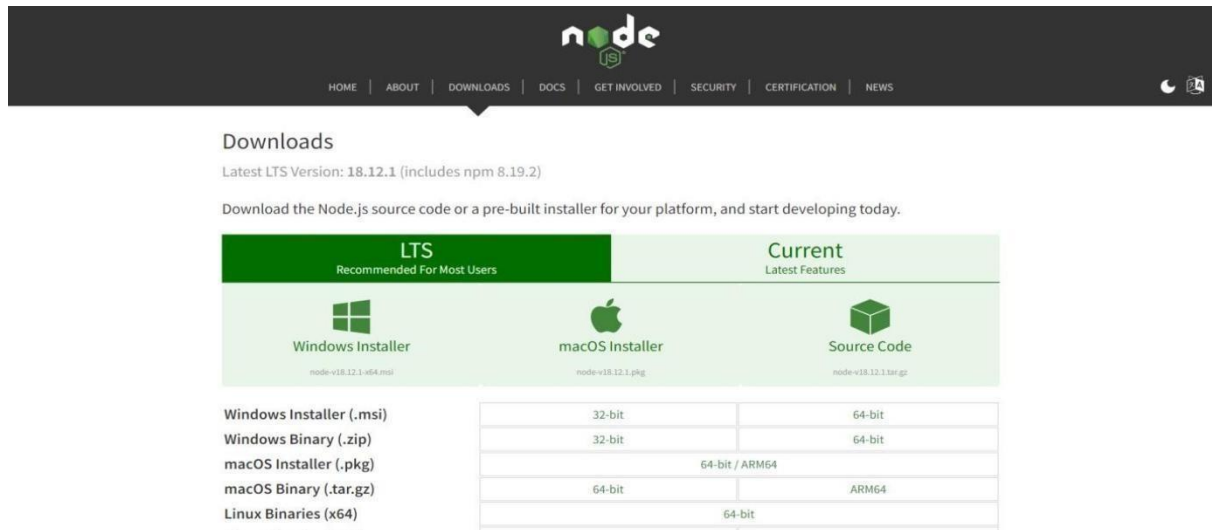
```
        print("Not connected to IoT")
        time.sleep(10)
```

```
deviceCli.commandCallback = myCommandCallback
```

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

NODE-JS CONNECTION:

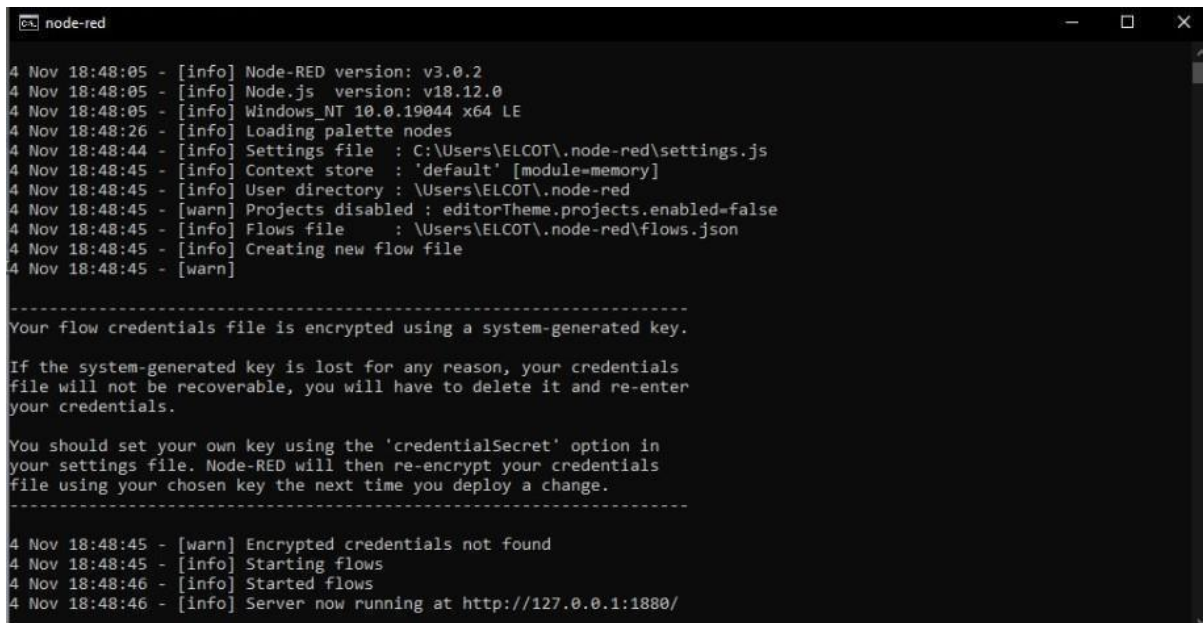
STEP1: Download and Install NODE JS.



The image shows the Node.js Downloads page. At the top, there's a navigation bar with links: HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, CERTIFICATION, and NEWS. Below the navigation bar, the 'Downloads' section is highlighted. It states 'Latest LTS Version: 18.12.1 (includes npm 8.19.2)' and 'Download the Node.js source code or a pre-built installer for your platform, and start developing today.' There are two main tabs: 'LTS Recommended For Most Users' and 'Current Latest Features'. Under the 'LTS' tab, there are three options: 'Windows Installer' (node-v18.12.1-x64.msi), 'macOS Installer' (node-v18.12.1.pkg), and 'Source Code' (node-v18.12.1.tar.gz). Below these, there are links for 'Windows Installer (.msi)', 'Windows Binary (.zip)', 'macOS Installer (.pkg)', 'macOS Binary (.tar.gz)', and 'Linux Binaries (x64)'. A table shows the available binaries for different architectures: 32-bit, 64-bit, 64-bit / ARM64, and ARM64.

Architecture	32-bit	64-bit	64-bit / ARM64	ARM64
32-bit	Available	Available		
64-bit		Available	Available	Available
64-bit / ARM64			Available	Available
ARM64				Available

STEP2: Setup node.js and configure command prompt for error check .open node-red from the generated link.



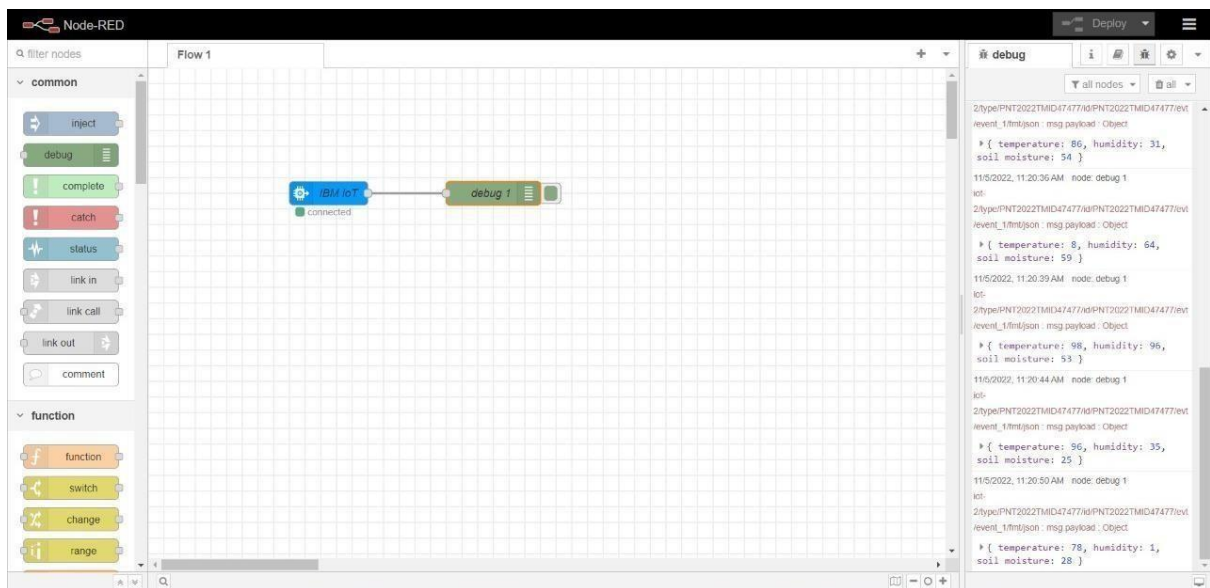
```
node-red
4 Nov 18:48:05 - [info] Node-RED version: v3.0.2
4 Nov 18:48:05 - [info] Node.js version: v18.12.0
4 Nov 18:48:05 - [info] Windows_NT 10.0.19044 x64 LE
4 Nov 18:48:26 - [info] Loading palette nodes
4 Nov 18:48:44 - [info] Settings file : C:\Users\ELCOT\.node-red\settings.js
4 Nov 18:48:45 - [info] Context store : 'default' [module=memory]
4 Nov 18:48:45 - [info] User directory : \Users\ELCOT\.node-red
4 Nov 18:48:45 - [warn] Projects disabled : editorTheme.projects.enabled=false
4 Nov 18:48:45 - [info] Flows file : \Users\ELCOT\.node-red\flows.json
4 Nov 18:48:45 - [info] Creating new flow file
4 Nov 18:48:45 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

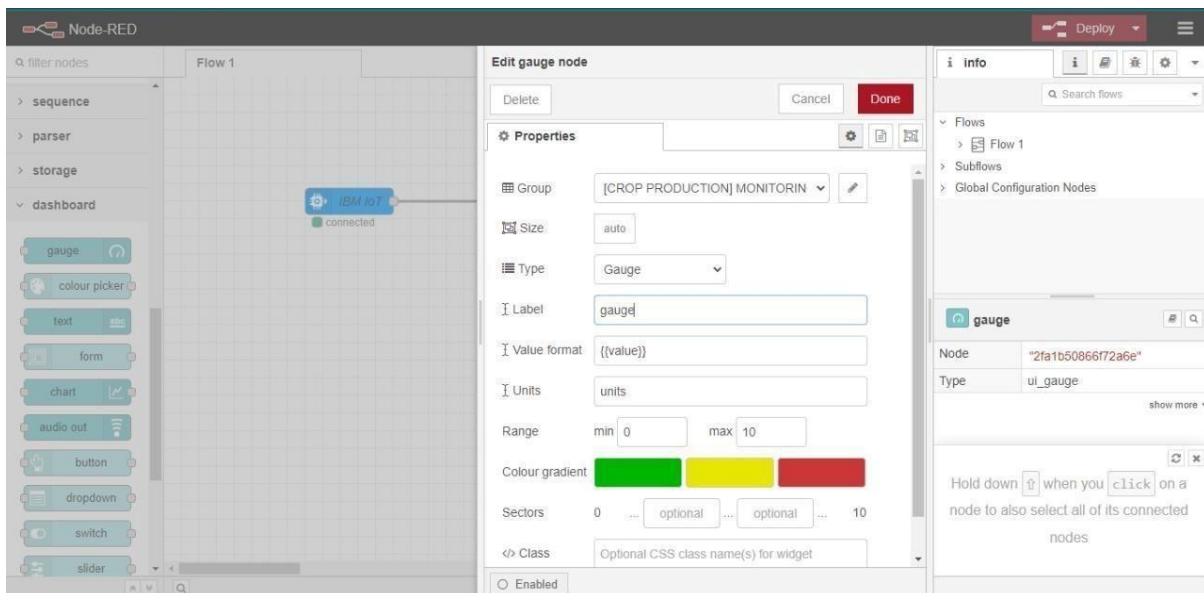
If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

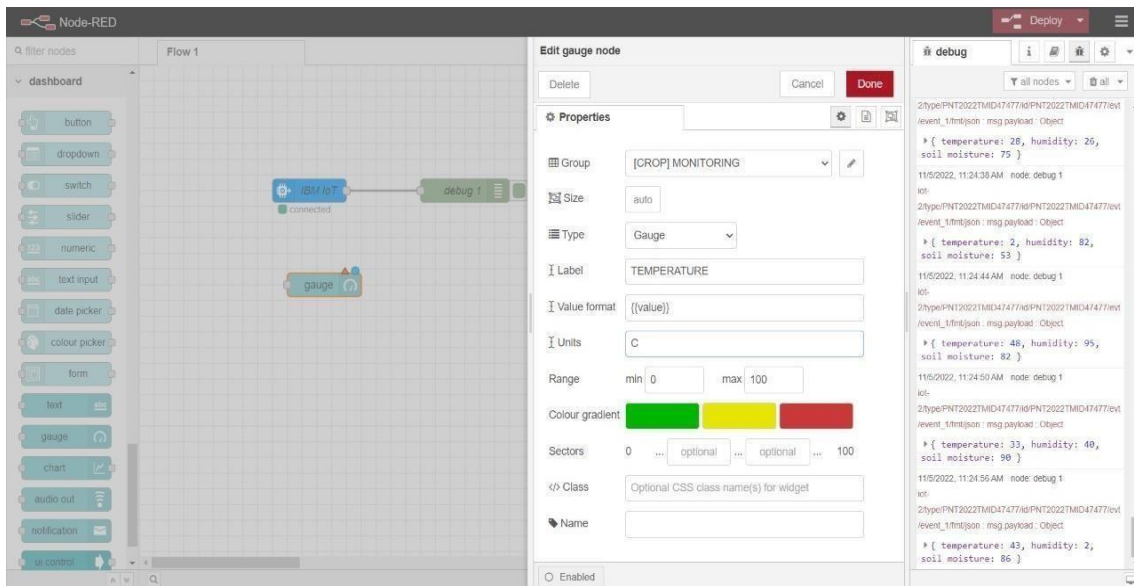
You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----
4 Nov 18:48:45 - [warn] Encrypted credentials not found
4 Nov 18:48:45 - [info] Starting flows
4 Nov 18:48:46 - [info] Started flows
4 Nov 18:48:46 - [info] Server now running at http://127.0.0.1:1880/
```

STEP3: Connect IBM IOT in and Debug 1 and Deploy.



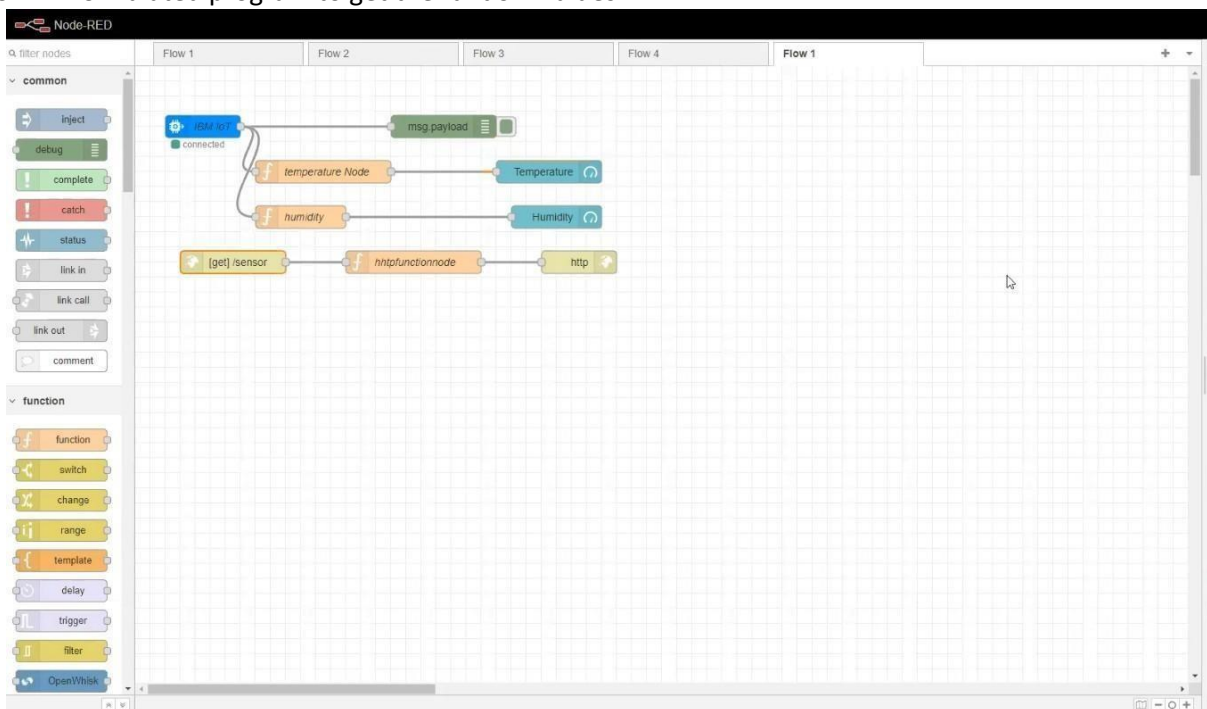
STEP4: Edit gauge node (Here the gauge nodes are named as Temperature, Humidity and Soilmoisture).



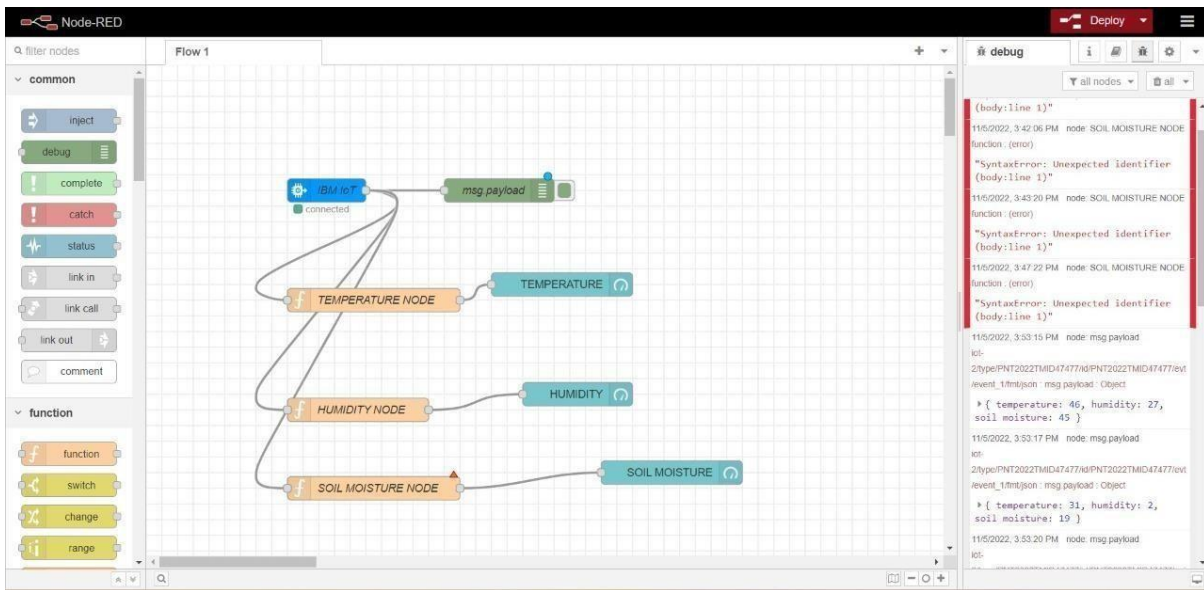


SIMULATION:

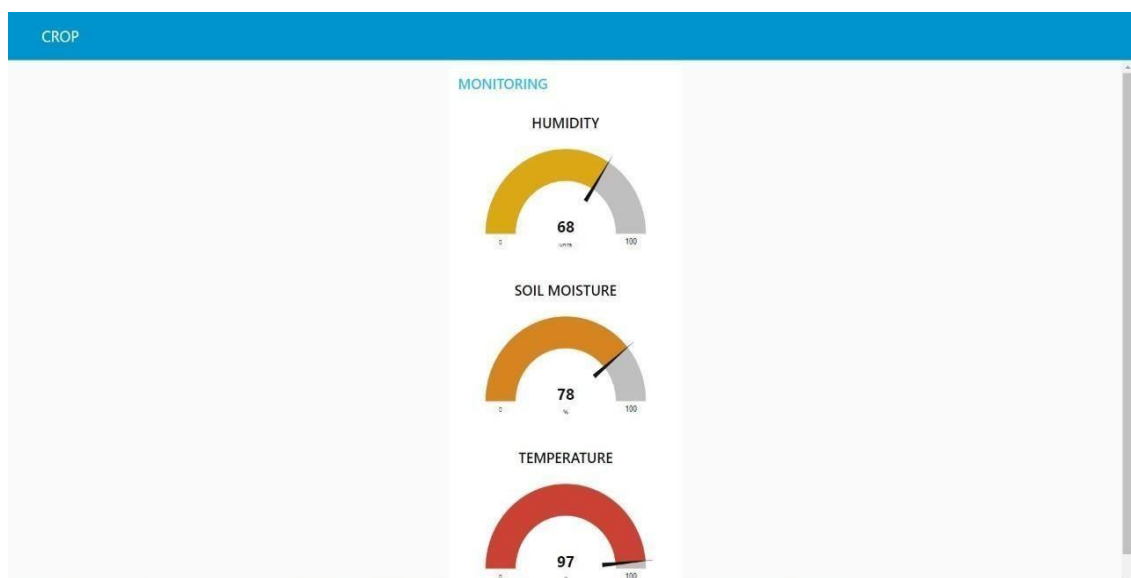
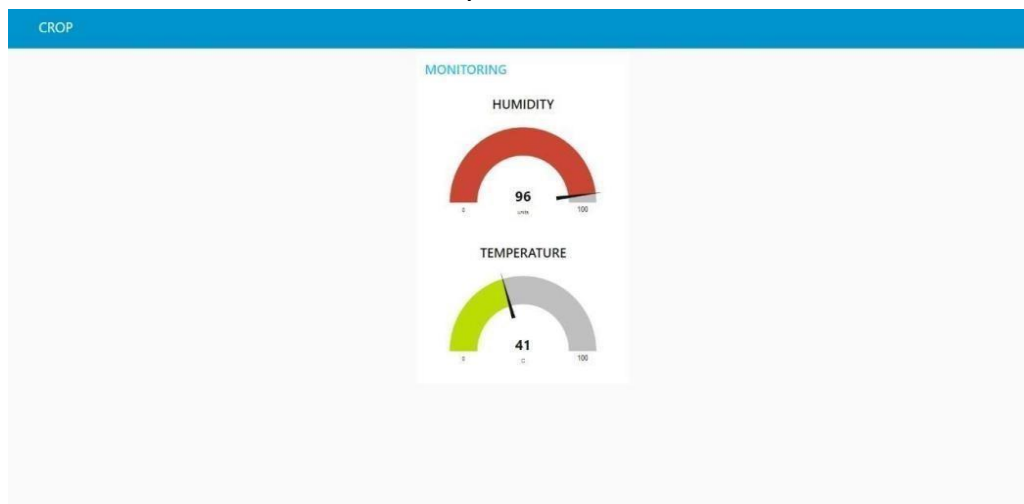
STEP1: Simulated program to get the random values



STEP2: Generate debug message from IBM Watson IoT Platform and connect the nodes.

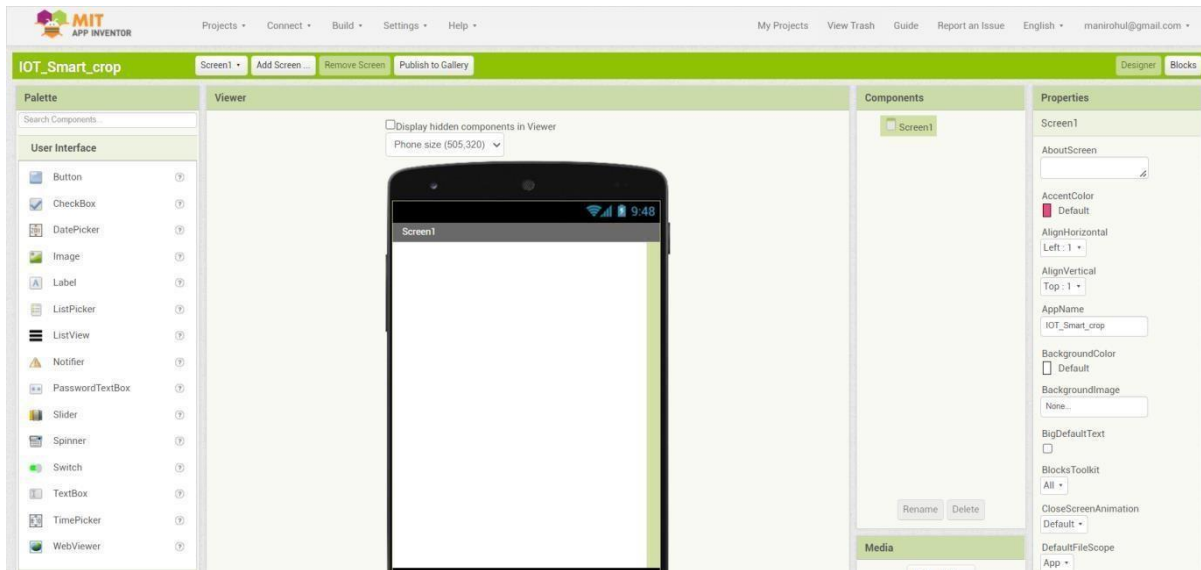


STEP3: Generate the some output from recent events.



MIT APP INVENTOR:

STEP 1: MIT APP inventor to design the APP.



STEP 2: Customize the App interface to Display the Values.

