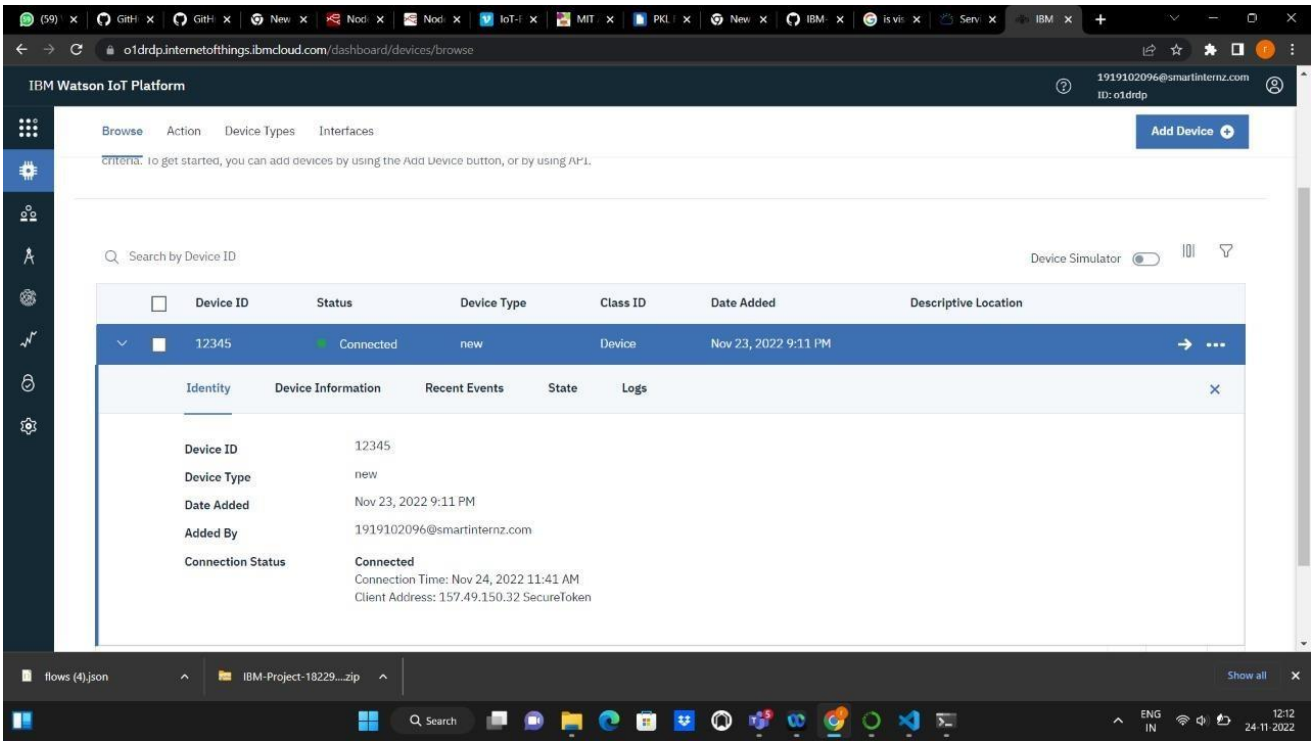
ponents & Technologies:

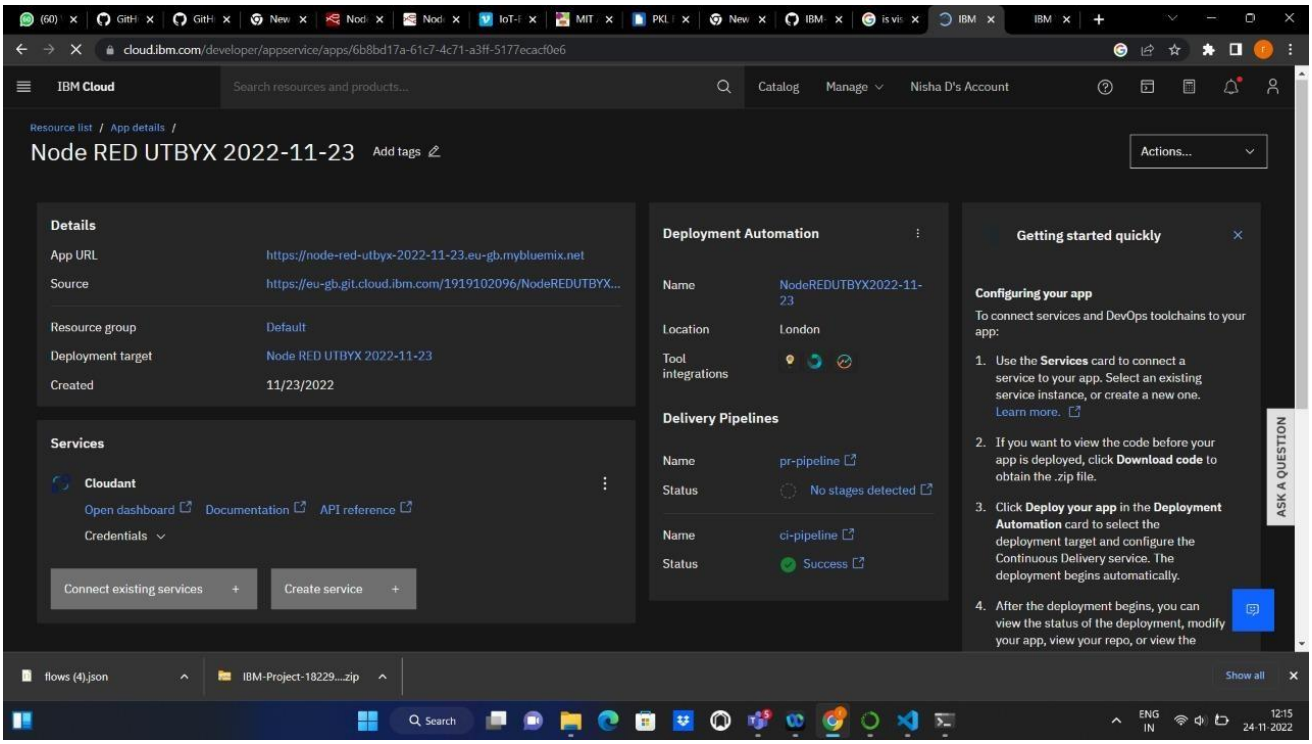
S.No	Component	Description	Technology
1	User Interface	How user interacts with application	HTML ,CSS, Javascript, React Js
2	Application Logic-1	Logic for the application	Python
3	Application Logic-2	Logic for the application	IBM Watson IOT service
4	Application Logic-3	Logic for the application	MySQL,NoSQL,etc.
5	Cloud Database	Database service on cloud	IBM Cloud
6	File Storage	It store the required files	IBM Block Storage or local filesystem
7	External API-1	Purpose of external application has been used in the application	IBM Weather API
8	Machine Learning Model	Purpose of machine learning model	Object recoginition model
9	Infrastructure	Local Server and cloud server configuration	Local, cloud foundry

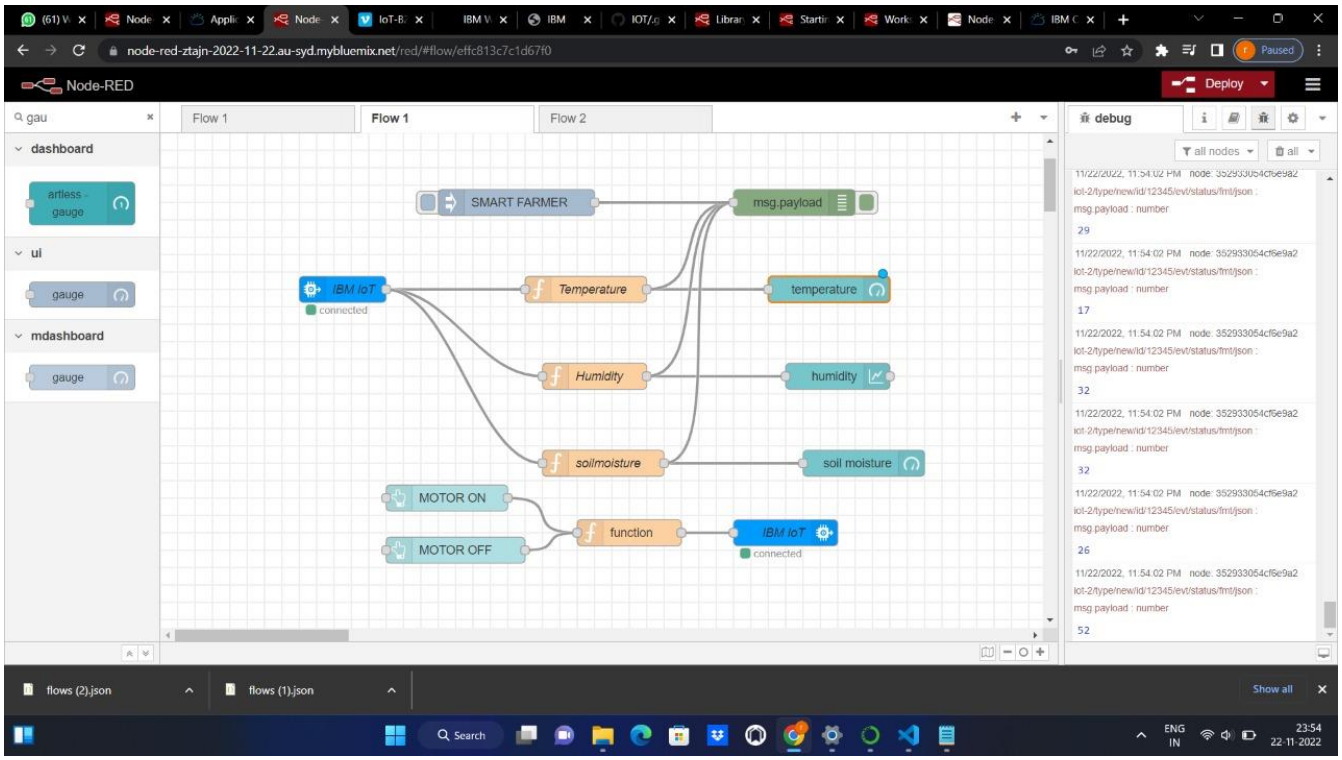
Application Characteristics:

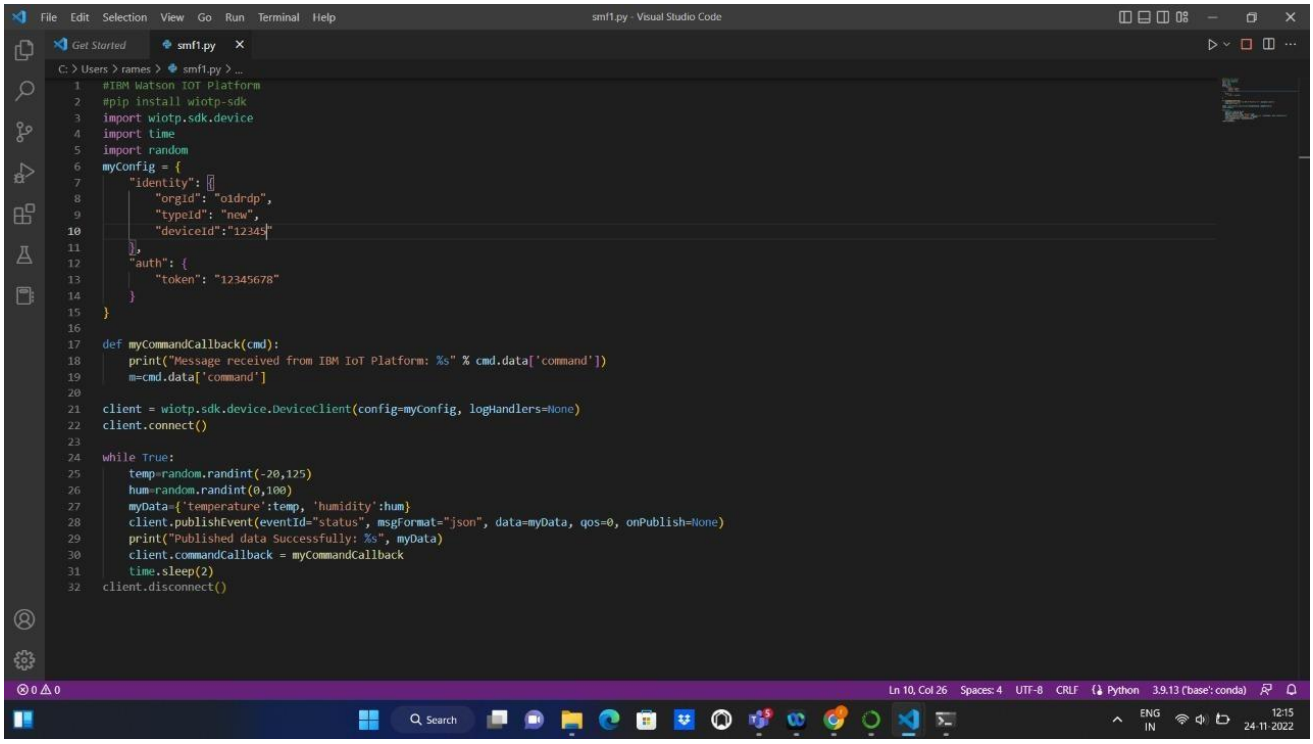
S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	List the frameworks used	Opensource frameworks
2	Security Implementations	Sensitive and private data must be protected from their production until the decision making and storage stages	Node red, open weather API, MIT app Inventor
3	Scalable Architecture	Scalability is a major concern for IOT platforms. It is choices of these platforms that would affect the scalability and that automatic real time decision making is feasible in an environment.	Technology used

APPLICATION PLATFORM AND CLOUD:






node-red-ztajn-2022-11-22.au-syd.mybluemix.net/red/#flow/effc813c7c1d67f0
Node-RED
Deploy
gau
dashboard
artless - gauge
ui
gauge
mdashboard
gauge
Flow 1
Flow 1
Flow 2
SMART FARMER
msg.payload
IBM IoT
connected
Temperature
temperature
Humidity
humidity
soilmoisture
soil moisture
MOTOR ON
MOTOR OFF
function
IBM IoT
connected
debug
all nodes
all
msg.payload : number
29
17
32
32
26
52
flows (2).json
flows (1).json
Show all
Search
ENG IN
23:54
22-11-2022


File Edit Selection View Go Run Terminal Help
smf1.py - Visual Studio Code
Get Started
smf1.py
C: > Users > rames > smf1.py > ...
#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
myConfig = {
"identity": {
"orgId": "o1drdp",
"typeId": "new",
"deviceId":"12345"
},
"auth": {
"token": "12345678"
}
}
def myCommandCallback(cmd):
print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
m=cmd.data['command']
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()
while True:
temp=random.randint(-20,125)
hum=random.randint(0,100)
myData={'temperature':temp, 'humidity':hum}
client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
print("Published data Successfully: %s", myData)
client.commandCallback = myCommandCallback
time.sleep(2)
client.disconnect()
Ln 10, Col 26 Spaces: 4 UTF-8 CRLF Python 3.9.13 ('base': conda)
Search
ENG IN
12:15
24-11-2022